

# **DETECTION AND CHARACTERIZATION OF DDOS ATTACKS USING TIME-BASED FEATURES**



Major Project submitted in partial fulfillment of the requirement for the award of the  
degree of

## **BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING**

Under the esteemed guidance of

**Miss A. ABHILASHA  
ASSISTANT PROFESSOR**

By

**BHOOPALAM KUSHAL - 19R11A05A0**



**Department of Computer Science and Engineering**  
**Accredited by NBA**

**Geethanjali College of Engineering and Technology**  
**(UGC Autonomous)**

(Affiliated to J.N.T.U.H, Approved by AICTE, New Delhi)  
Cheeryal (V), Keesara (M), Medchal.Dist.-501 301.

**May-2023**

# Geethanjali College of Engineering & Technology

(UGC Autonomous)

(Affiliated to JNTUH, Approved by AICTE, New Delhi)

Cheeryal (V), Keesara(M), Medchal Dist.-501 301.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Accredited by NBA



## CERTIFICATE

This is to certify that the B.Tech Major Project report entitled “ **DETECTION AND CHARACTERIZATION OF DDOS ATTACKS USING TIME-BASED FEATURES** ” is a bonafide work done by **BETHI VISHWAMBHAR (19R11A05A0)**, in partial fulfillment of the requirement of the award for the degree of Bachelor of Technology in “**Computer Science and Engineering**” from Jawaharlal Nehru Technological University, Hyderabad during the year 2022-2023.

**Internal Guide**

**Miss A. Abhilasha**

**Assistant Professor**

**HOD - CSE**

**Dr. A. Sree Lakshmi**

**Professor**

**External Examiner**

# Geethanjali College of Engineering & Technology

(UGC Autonomous)

(Affiliated to JNTUH Approved by AICTE, New Delhi)  
Cheeryal (V), Keesara(M), Medchal Dist.-501 301.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Accredited by NBA



## DECLARATION BY THE CANDIDATE

I, **BHOOPALAM KUSHAL**, bearing Roll No. **19R11A05A0**, hereby declare that the project report entitled “**DETECTION AND CHARACTERIZATION OF DDOS ATTACKS USING TIME-BASED FEATURES**” is done under the guidance of **Miss A. Abhilasha** , **Assistant Professor**, Department of Computer Science and Engineering, Geethanjali College of Engineering and Technology, is submitted in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering**.

This is a record of bonafide work carried out by me/us in **GEETHANJALI COLLEGE OF ENGINEERING AND TECHNOLOGY** and the results embodied in this project have not been reproduced or copied from any source. The results embodied in this project report have not been submitted to any other University or Institute for the award of any other degree or diploma.

**BHOOPALAM KUSHAL – (19R11A05A0)**

**Department of CSE,**

**Geethanjali College of Engineering and Technology, Cheeryal.**

# ACKNOWLEDGEMENT

I am greatly indebted to the Management of Geethanjali College of Engineering and Technology, Cheeryal , Hyderabad, for providing us the necessary facilities to successfully carry out this major project work titled “**DETECTION AND CHARACTERIZATION OF DDOS ATTACKS USING TIME-BASED FEATURES**”.

Firstly, I thank and express my sincere gratitude to **Professor Dr. A. Sree Lakshmi, CSE department, Geethanjali College of Engineering and Technology** for her invaluable help and support which helped me a lot in successfully completing my project.

Moreover, I also express my gratitude to **Assistant Professor A. ABHILASHA, Geethanjali College of Engineering and Technology** , my guide and patron, for his continued support throughout my endeavor to make my project successfully done.

I would like to express my sincere gratitude to our **Principal Dr. S. UDAYA KUMAR** for providing the necessary infrastructure to complete my project.

I convey my gratitude to our **Chairman, Mr. G. RAVINDER REDDY**, for his invaluable support and encouragement for propelling my innovation forward.

Finally, I would like to express my heartfelt gratitude to my parents and all my peers who were very supportive and for their encouragement to achieve my goals.

With regards,

**BHOOPALAM KUSHAL - 19R11A05A0**

**Department of CSE,**

**Geethanjali College of Engineering and Technology, Cheeryal.**

# ABSTRACT

A Distributed Denial-of-Service (DDoS) attack is an attack meant to shut down a machine or network, making it inaccessible to its intended users. DDOS attacks accomplish this by flooding the target with traffic, or sending it information that triggers a crash. This project aims at analyzing the DDoS attacks data set by using Gaussian Naive Bayes Method and Adaboost Classification method. The correlated columns of the data set are identified by generating the heat map, and those columns are dropped to reduce the unnecessary load on the data set. The project comprises of four modules. The first module deals with the identification of correlated columns by generating the heatmap. The second module deals with the Gaussian Naive Bayes analysis. The third module deals with the Adaboost Classification analysis. The fourth module deals with the generation of a comparison plot between the Gaussian Naïve Bayes and Adaboost Classifier..

## LIST OF FIGURES

Figure No.	Figure Description	Page Number
1	System Architecture	15
2	Use Case Diagram	19
3	Activity Diagram	20
4	Sequence Diagram	21
5	Basic Levels of Testing	46

## LIST OF ABBREVIATIONS

S. No	Abbreviations	Full Form
1	UML	Unified Modelling Language
2	DDoS	Distributed-Denial-of-Service
3	GNB	Gaussian Naive Bayes
4	ABC	Adaboost Classification
5	CSV	Comma Separated Value

## LIST OF IMAGES

S. No	Name	Page No
1	Anaconda Prompt	49
2	Home Screen	50
3	Data Columns Filtering	51
4	CSV File	52
5	Gaussian Naive Bayes Analysis	53
6	Accuracy of GNB	54
7	Adaboost Analysis	55
8	Accuracy of ABC	56
9	Comparison Plot	57
10	Comparison Graph	58
11	Heatmap Generation	59
12	Heatmap Plot	60

# TABLE OF CONTENTS

<b>S. No</b>	<b>Contents</b>	<b>Page no</b>
i	Abstract	v
ii	List of Figures	vi
iii	List of Tables	vi
iv	List of Images	vii
<b>1.</b>	<b>Introduction</b>	<b>1</b>
	1.1 About the project	
	1.2 Objective	
<b>2.</b>	<b>System Analysis</b>	<b>3</b>
	2.1 Existing System	
	2.2 Proposed System	
	2.2.1 Details	
	2.2.2 Ethics	
	2.3 Software Requirement Specification	
	2.4 Scope of the Project	
	2.5 System Configuration	
	2.6 Feasibility Analysis	
<b>3.</b>	<b>Literature Overview</b>	<b>9</b>
	3.1 Project Literature	
	3.2 Technologies	
	3.3 Survey	



<b>4.</b>	<b>System Design</b>	<b>15</b>
	4.1 System Architecture	
	4.2 UML Diagram	
	4.3 Use Case Diagram	
	4.4 Activity Diagram	
	4.5 Sequence Diagram	
<b>5.</b>	<b>Sample Code</b>	<b>22</b>
	5.1 Coding	
<b>6.</b>	<b>Testing</b>	<b>45</b>
	6.1 Testing	
	6.2 Test cases	
<b>7.</b>	<b>Output Screens</b>	<b>49</b>
<b>8.</b>	<b>Conclusion</b>	<b>61</b>
	8.1 Conclusion	
	8.2 Further Enhancements	
<b>9.</b>	<b>Bibliography</b>	<b>63</b>
	9.1 Books References	
	9.2 Websites References	
	9.3 Technical Publication References.	
<b>10</b>	<b>Appendices</b>	<b>65</b>
	10.1. Software Used	
	10.2. Testing Methods	
<b>11</b>	<b>Plagiarism Report</b>	<b>67</b>

# 1.INTRODUCTION

## 1.1 About the Project

This project aims at analyzing the DDoS attacks data set by using Gaussian Naive Bayes Method and Adaboost Classification method. The correlated columns of the data set are identified by generating the heat map, and those columns are dropped to reduce the unnecessary load on the data set.

The project comprises of four modules. The first module deals with the identification of correlated columns by generating the heatmap. The second module deals with the Gaussian Naive Bayes analysis. The third module deals with the Adaboost Classification analysis. The fourth module deals with the generation of a comparison plot between the Gaussian Naive Bayes and Adaboost Classification.

A Naive Bayes classifier is a probabilistic machine learning model that's used for classification task by using Baye's theorem of probability. Naive Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem. A closed form expression is a mathematical expression that can be evaluated in a finite number of operations. It may contain constants, variables, certain "well-known operations and functions. Maximum likelihood training can be done by evaluating a closed-form expression, which takes linear time, rather than by expensive iterative approximation as used for many other types of classifiers.

**Tools Used:** Sklearn's Gaussian Naive Bayes & Adaboost Classification Outputs from the project:

Heatmap plot, Gaussian Naive Bayes Accuracy Score, Adaboost Classification Accuracy Score and Comparison Plot In the existing system correction measures are initiated after the DDOS Attack happens. This project propose to predict DDOS attack so that precautionary measures can be planned before the attack.

## 1.2 Objectives

### 1.2.1 Admin

### 1.2.2. User

#### 1.2.1 Admin Module

Admin is the super user of the GUI who can manage everything on the GUI.

- a. Dashboard:** In this section, the admin can see all details in brief like the total, assigned and the sample collected and completed tests.
- b. List of Modules:** In this section, the user can see all the modules including Dataset Columns filtering, Gaussian Naive Bayes Analysis, Adaboost Analysis, Comparison Plot and Heat map generation.

#### User Module

User can visit GUI through the command prompt.

- a. Selection:** User can select any desired module to see the output of each module.
- b. Modules:**
  - i. **Dataset Columns Filtering:** Here the correlated columns get dropped to reduce the unnecessary load on the dataset and processed csv dataset is generated.
  - ii. **Gaussian Naive Bayes:** The user can able to see the Gaussian Naive Bayes accuracy score in the command prompt by just clicking the button.
  - iii. **Adaboost Analysis:** The user can able to see the Adaboost Analysis accuracy score in command prompt.
  - iv. **Comparison Plot:** The Comparison Plot between Gaussian Naive Bayes Analysis and Adaboost Analysis can be seen in the GUI.
  - v. **Heat Map Generation:** Here the correlated columns are identified by generating the heatmap and can be seen in the GUI.

## 2.SYSTEM ANALYSIS

### 2.1 Existing System

In DDoS attacks, there is one server and there are multiple systems trying to throw invalid requests simultaneously. Even after the server responds to this request, the system throws the same request repeatedly. This results in the crashing of the server. In the existing system correction measures are initiated after the DDOS Attack happens.

#### DISADVANTAGES

- a. **Reactive Approach:** The existing system takes a reactive approach to DDoS attacks, meaning that countermeasures are only taken after the attack has already caused damage. This can lead to a significant loss of resources, data, and reputation.
- b. **Time Delay:** The response time of the existing system can be slow, which means that the damage can already be done by the time countermeasures are initiated. This can result in extended downtime, which can affect business operations and result in financial losses.

### 2.2 Proposed System

This project proposes to predict DDOS attack so that precautionary measures can be planned before the attack. Along with precautionary measures Heatmap plot, Gaussian Naive Bayes Accuracy Score, Adaboost Classification Accuracy Score, Comparison Plots are also produced.

#### ADVANTAGES

- a. **Proactive Approach:** By predicting DDoS attacks, organizations can take a proactive approach to prevent them from happening in the first place. This can significantly reduce the risk of damage to resources, data, and reputation.

- b. Time Savings:** Predicting DDoS attacks can save time by allowing organizations to take preventive measures before the attack occurs, rather than responding to it after the fact. This can help minimize downtime and reduce the impact on business operations.

### **2.2.1 Details**

A Distributed Denial-of-Service (DDoS) attack is a malicious cyber-attack that targets a single system or network, making it unavailable to users. It is carried out by overwhelming the target system or network with a flood of traffic from multiple sources, thereby denying legitimate users access to the system or network resources.

### **2.2.2 Ethics**

- a.** DDoS attacks are unethical because they disrupt the normal functioning of systems and cause harm to organizations and their customers.
- b.** Collateral damage is a significant ethical consideration in DDoS attacks, as innocent parties can be affected.
- c.** DDoS attacks are illegal in most countries and can result in legal consequences for those who carry them out.
- d.** Ethical responsibility requires individuals and organizations to respect the privacy and security of others and avoid actions that may cause harm or disruption.
- e.** Ethical hacking can be a valuable tool for organizations to test their security systems and identify vulnerabilities before they can be exploited by malicious actors.

## **2.3 Software Requirement Specification**

The SRS document outlines the functional and non-functional requirements of the software, as well as any constraints or assumptions that may affect the software development process. It serves as a communication tool between the stakeholders, including the customers, developers, testers, and project managers.

### **2.3.1 Functional requirements:**

The project needs the requirement of few technologies like PyQt Designer, SQLite, Pyuic etc for the proper functioning of the system.

### **2.3.2 Non-functional requirements:**

The project is very reliable as per specified functions. It is very safe and secure to use this to know the accuracy score of each algorithm based on dataset and it is easy to maintain.

### **2.3.3 User Interface:**

The user interacts with the system to experience the interface and also to know about the different elements present in the project simply by just clicking on the required buttons.

### **2.3.4 Dependencies:**

The project basically depends on the dataset and based on the dataset values and its parameters it produces the accuracy score for each algorithm.

### **2.3.5 Constraints:**

Using algorithms other than Gaussian Naive Bayes and Adaboost Analysis may not generate optimal accuracy score.

## **2.4 Scope of the Project**

The project scope of DDoS (Distributed Denial of Service) attacks detection and prevention involves implementing measures to identify and mitigate attacks that attempt to overwhelm a network or server with excessive traffic. This includes:

- a.** Monitoring network traffic and identifying anomalous patterns that indicate a DDoS attack.
- b.** Using firewalls, load balancers, and intrusion detection and prevention systems to mitigate attacks.
- c.** Employing rate limiting and traffic filtering techniques to block malicious traffic and allow legitimate traffic.
- d.** Implementing failover and backup systems to ensure availability of critical services in the event of an attack.

### **Advantages:**

- a.** Improved uptime and availability of critical services, reducing the risk of lost revenue and customer dissatisfaction.
- b.** Reduced risk of data breaches and loss of sensitive information.
- c.** Improved customer confidence and trust in the security of the organization.
- d.** Compliance with regulatory requirements and industry best practices.
- e.** Cost savings by avoiding the need for expensive incident response and recovery efforts.
- f.** Improved reputation and brand image by demonstrating a commitment to security and resilience.

## 2.5 System Configuration

### 2.5.1 HARDWARE REQUIREMENTS

**a. PROCESSOR:** Any Quadcore Processor

**b. RAM:** 8GB

### 2.5.2 SOFTWARE REQUIREMENTS

**a. Operating System:** Windows 7 or later, macOS and Linux.

**b. Front End:** Python Qt Designer, Pyuic, Python.

**c. Back End:** SQLite3

## 2.6 Feasibility Analysis

As the name implies, a feasibility study is used to determine the viability of an idea, such as ensuring a project is legally and technically feasible as well as economically justifiable. It tells us whether a project is worth the investment—in some cases, a project may not be doable. There can be many reasons for this, including requiring too many resources, which not only prevents those resources from performing other tasks but also may cost more than an organization would learn back by taking on a project that isn't profitable.

### 2.6.1 Economical Feasibility

This assessment typically involves a cost/ benefits analysis of the project, helping organizations determine the viability, cost, and benefits associated with a project before financial resources are allocated. It also serves as an independent project assessment and enhances project credibility—helping decision makers determine the positive economic benefits to the organization that the proposed project will provide. Our project is economically feasible because in this we have used “UBUNTU”, “PYTHON”, “PYQT” designer tool and “PYUIC” which are all available as an open source.



### **2.6.2 Technical Feasibility**

This assessment focuses on the technical resources available to the organization. It helps organizations determine whether the technical resources meet capacity. Technical feasibility also involves evaluation of the hardware, software, and other technology requirements of the proposed system. A prototype of the tool was developed to verify the technical feasibility. The prototype is working successfully and hence the project is feasible.

## **3.LITERATURE OVERVIEW**

### **3.1 Project Literature**

Literature survey is the most important step in software development process. Before developing the tool, it is necessary to determine the time factor, economy and company strength. Once these things are satisfied, then next step is to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from book or from websites.

In this project, two machine learning algorithms, Gaussian Naive Bayes and Adaboost Classification, are applied to the DDoS attacks dataset. The aim is to compare the performance of these algorithms in detecting DDoS attacks and to generate a comparison plot between them. Before applying the algorithms, correlated columns in the dataset are identified using a heatmap, and those columns are dropped to reduce the load on the dataset.

#### **3.1.1 Gaussian Naive Bayes:**

Gaussian Naive Bayes is a probabilistic classifier that uses Bayes' theorem of probability to assign a probability of a connection being an attack or normal. It is a highly scalable algorithm that requires a small number of parameters, making it a suitable choice for large datasets. GNB assumes that the features of a data point are independent and normally distributed. The algorithm calculates the likelihood of a data point belonging to a particular class based on the probability density function of each feature. Then, it uses Bayes' theorem to calculate the posterior probability of the class given the data point. In summary, GNB is a probabilistic algorithm that calculates the probability of a data point belonging to a particular class based on the probability density function of each feature. It is widely used in text classification, spam filtering, and other classification tasks.

### **3.1.2 Adaboost Classification:**

Adaboost Classification is an ensemble learning algorithm that combines multiple weak classifiers to build a strong classifier. It iteratively trains weak classifiers on different subsets of the dataset, and then combines them to form a final model. Adaboost works by iteratively training a series of "weak" classifiers on different subsets of the training data, and then combining their outputs in a weighted fashion to form a "strong" classifier. In each iteration, Adaboost places greater emphasis on the training examples that were misclassified by the previous set of weak classifiers, effectively adapting to the complexity of the data and improving the overall accuracy of the model.

In addition to the two algorithms, the project also produces a heatmap plot, Gaussian Naive Bayes accuracy score, Adaboost Classification accuracy score, and a comparison plot. These outputs provide valuable insights into the performance of the algorithms and can aid in selecting the most appropriate algorithm for a given scenario.

Overall, the project demonstrates the usefulness of machine learning algorithms in detecting DDoS attacks and highlights the importance of selecting an appropriate algorithm based on the characteristics of the dataset and the application requirements. It also emphasizes the need for proactive measures to prevent DDoS attacks rather than relying solely on reactive measures after an attack has occurred.

## **3.2 Technologies**

### **3.2.1 Python**

Python is an interpreted, high-level programming language that was first released in 1991. It is designed to be easy to learn and read, which makes it a popular choice for beginners and experienced programmers alike. Python is used for a wide range of applications, including scientific computing, data analysis, web development, machine learning, and more. Some of the key features of Python include its simple syntax, dynamic typing, and built-in support for common programming tasks such as file I/O and regular expressions.

### **3.2.2 PyQt Designer**

PyQt Designer is a visual interface design tool that is used to create user interfaces for PyQt applications. It is a drag-and-drop interface that allows you to place widgets and other UI elements on a form, and then customize their properties using a properties editor. PyQt Designer supports a wide range of widgets and layout managers, making it easy to create complex and professional-looking interfaces. Once you have designed your interface, you can save it as a .ui file, which can be converted to Python code using Pyuic.

### **3.2.3 SQLite3**

SQLite3 is a lightweight, file-based relational database management system that is used to store and retrieve data. It is often used in small-scale applications and mobile apps because of its simplicity and ease of use. SQLite3 stores data in a single file, which makes it easy to distribute and deploy. It supports a wide range of data types, including text, integer, real, blob, and null. SQLite3 is also very fast and efficient, which makes it a good choice for applications that need to work with large amounts of data.

### **3.2.4 Pyuic**

Pyuic is a command-line tool that is used to convert PyQt Designer .ui files to Python code. It is a utility tool that comes with the PyQt library, and it takes the .ui file created in PyQt Designer and generates the corresponding Python code. The generated code can be used directly in the application, which saves time and effort in coding the user interface. Pyuic supports a wide range of PyQt widgets and layout managers, making it easy to convert even complex interfaces to Python code.

### 3.3 Survey

#### 1. A SURVEY OF DISTRIBUTED DENIAL-OF-SERVICE ATTACK, PREVENTION, AND MITIGATION TECHNIQUES (IJCRT)

**Authors:** A. Jagekar, S.K, Jadhav.

Distributed denial-of-service is one kind of the most highlighted and most important attacks of today's cyberworld. With simple but extremely powerful attack mechanisms, it introduces an immense threat to current Internet community. In this article, we present a comprehensive survey of distributed denial-of-service attack, prevention, and mitigation techniques. We provide a systematic analysis of this type of attacks including motivations and evolution, analysis of different attacks so far, protection techniques and mitigation techniques, and possible limitations and challenges of existing research. Finally, some important research directions are outlined which require more attentions in near future to ensure successful defense against distributed denial-of-service attacks.

The flood of incoming messages, connection requests or malformed packets to the target system forces it to slow down or even crash and shut down, thereby denying service to legitimate users or systems. This paper presents a literature review of DDoS attacks and the common defense mechanisms available. It also presents a literature review of the defenses for low-rate DDoS attacks that have not been handled effectively hitherto.

In general, DDoS attack packets do not show any obvious characteristics which can separate a malicious stream from a legitimate one. Also, the tools used in these attacks are easily accessible by the attackers and this increases the frequency and threats of the attacks. The simple structure of this kind of attacks follows many-to-one feature.

This limitation introduces a deep trouble known as IP spoofing which is one of the key powers of the DDoS attacks.<sup>4</sup> In IP spoofing, the attackers provide false information such as fake source IP addresses in the IP packets.<sup>8</sup> Furthermore, the routers also cannot provide packet tracing mechanisms because of the enormous traffic handled by them.

## **2. A SURVEY ON INTERNET TRAFFIC IDENTIFICATION BY IEEE COMMUN. SURVEYS TUTS.**

**Authors:** Mirchev, M.J., Mirtchev, S.T. (2020).

The area of Internet traffic measurement has advanced enormously over the last couple of years. This was mostly due to the increase in network access speeds, due to the appearance of bandwidth-hungry applications, due to the ISPs' increased interest in precise user traffic profile information and also a response to the enormous growth in the number of connected users. These changes greatly affected the work of Internet service providers and network administrators, which have to deal with increasing resource demands and abrupt traffic changes brought by new applications. This survey explains the main techniques and problems known in the field of IP traffic analysis and focuses on application detection.

The area of Internet traffic measurement has advanced enormously over the last couple of years. This was mostly due to the increase in network access speeds, due to the appearance of bandwidth-hungry applications, due to the ISPs' increased interest in precise user traffic profile information and also a response to the enormous growth in the number of connected users. These changes greatly affected the work of Internet service providers and network administrators, which have to deal with increasing resource demands and abrupt traffic changes brought by new applications.

This survey explains the main techniques and problems known in the field of IP traffic analysis and focuses on application detection. First, it separates traffic analysis into packet-based and flow-based categories and details the advantages and problems for each approach. Second, this work cites the techniques for traffic analysis accessible in the literature, along with the analysis performed by the authors. Relevant techniques include signature-matching, sampling and inference. Third, this work shows the trends in application classification analysis and presents important and recent references in the subject. Lastly, this survey draws the readers' interest to open research topics in the area of traffic analysis and application detection and makes some final remarks.

### **3. REAL-TIME PERFORMANCE ANALYSIS ON DDOS ATTACK DETECTION USING MACHINE LEARNING BY BANGLADESH ASSOCIATION OF COMMONWEALTH UNIVERSITY.**

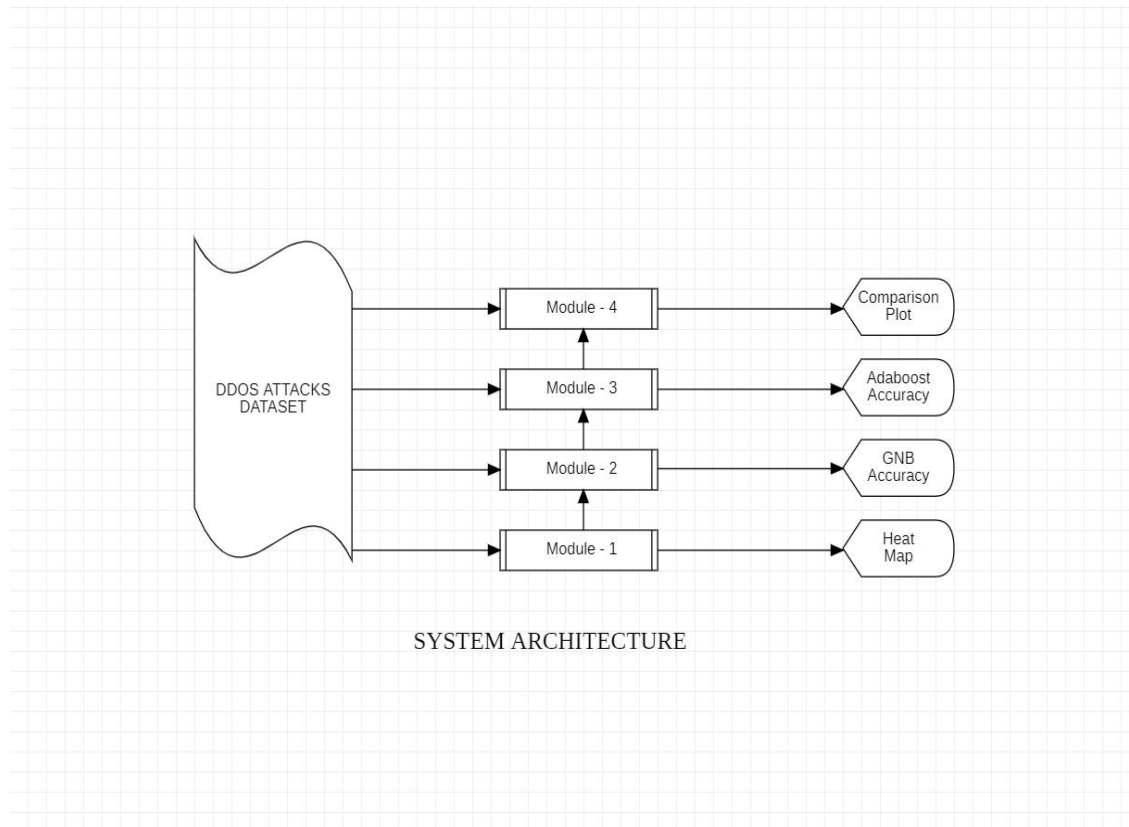
**Authors:** Suvra, D.K.S. Sen, T. Hossain, M.I. Rahman, A. Mou, M.M. (2020).

In recent years, Distributed Denial of service (DDoS) attacks have led to a tremendous financial loss in some industries and governments. Such as banks, universities, news and media publications, financial services, political or governmental servers. DDoS attack is one of the biggest threats for cyber security nowadays. It is a malicious act that slows down the server, makes loss of confidential data and makes reputation damage to a brand. With the advancement of developing technologies for example cloud computing, Internet of things (IoT), Artificial intelligence attackers can launch attacks very easily with lower cost. However, it is challenging to detect DDoS traffic as it is similar to normal traffic.

In this era, we rely on the internet services. Attackers send a huge volume of traffic at the same time to a specific network and make the network null and void. So that the server cannot respond to the actual users. As a result, clients cannot get the services from that server. It is very essential to detect DDoS attacks and secure servers from losing important information and data. However, many detection techniques are available for preventing the attack. But it is very challenging to choose one method among those as some are time efficient and some are result oriented. In our paper, we mainly focused on the top machine learning classification algorithms and evaluated the best model according to the dataset. The experimental result shows that the Decision Tree algorithm achieved the excellent accuracy of 98.50 percent with very less time consumption. Therefore, we are using a better approach to detect DDoS attacks in real time.

## 4.SYSTEM DESIGN

### 4.1 System Architecture



**Figure: 4.1 System Architecture**

A software system's architecture typically includes different levels of abstraction, from high-level conceptual designs to detailed low-level technical specifications. The architecture is usually described using various diagrams, such as block diagrams, flowcharts, and UML diagrams, which represent the system's structure and behavior.

The system architecture plays a critical role in software development, as it provides a blueprint for the development team to follow. It helps ensure that the system meets its functional and non-functional requirements, such as reliability, scalability,



performance, and maintainability. A well-designed system architecture can also make the system easier to understand, modify, and extend over time.

#### **4.1.1 The system architecture can be divided into four main modules:**

##### **Module 1: HeatMap Plot**

The first module deals with the identification of correlated columns by generating the heatmap. The axis variables are divided into ranges like a bar chart or histogram, and each cell's color indicates the value of the main variable in the corresponding cell range.

##### **Module 2: Gaussian Naive Bayes Accuracy**

The second module deals with the Gaussian Naive Bayes analysis. It is a probabilistic classifier, which means it predicts on the basis of the probability of an object.

##### **Module 3: Adaboost Accuracy**

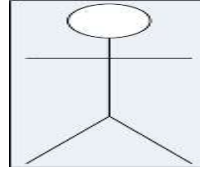
The third module deals with the Adaboost Classification analysis. It is called Adaptive Boosting as the weights are re-assigned to each instance, with higher weights assigned to incorrectly classified instances.

##### **Module 4: Comparison Plot**

The fourth module deals with the generation of a comparison plot between the Gaussian Naive Bayes and Adaboost Classification.

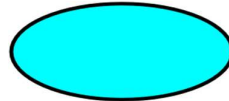
## 4.2 UML Diagrams

- a. Actor:** A coherent set of roles that users of use cases play when interacting with the use cases.



**Figure: a**

- b. Use Case:** A description of sequence of actions, including variants, that a system performs that yields an observable result of value of an actor.



**Figure: b**

UML stands for Unified Modeling Language. UML is a language for specifying, visualizing and documenting the system. This is the step while developing any product after analysis. The goal from this is to produce a model of the entities involved in the project which later need to be built. The representation of the entities that are to be used in the product being developed need to be designed.

UML stands for Unified Modeling Language, which is a standardized graphical language used to represent software systems in software engineering. UML diagrams are a set of graphical notations that allow software developers to communicate and visualize different aspects of a software system's design and behavior.

UML diagrams are used throughout the software development process, from initial requirements gathering and design to implementation and testing. They provide a standardized way for software developers, designers, and stakeholders to communicate and understand the different aspects of a software system.

## 4.3 USECASE DIAGRAMS

Use case diagrams model behavior within a system and helps the developers understand of what the user require. The stick man represents what's called an actor. Use case diagram can be useful for getting an overall view of the system and clarifying that can do and more importantly what they can't do. Use case diagram consists of use cases and actors and shows the interaction between the use case and actors.

- a. The purpose is to show the interactions between the use case and actor.
- b. To represent the system requirements from user's perspective.
- c. An actor could be the end-user of the system or an external system.

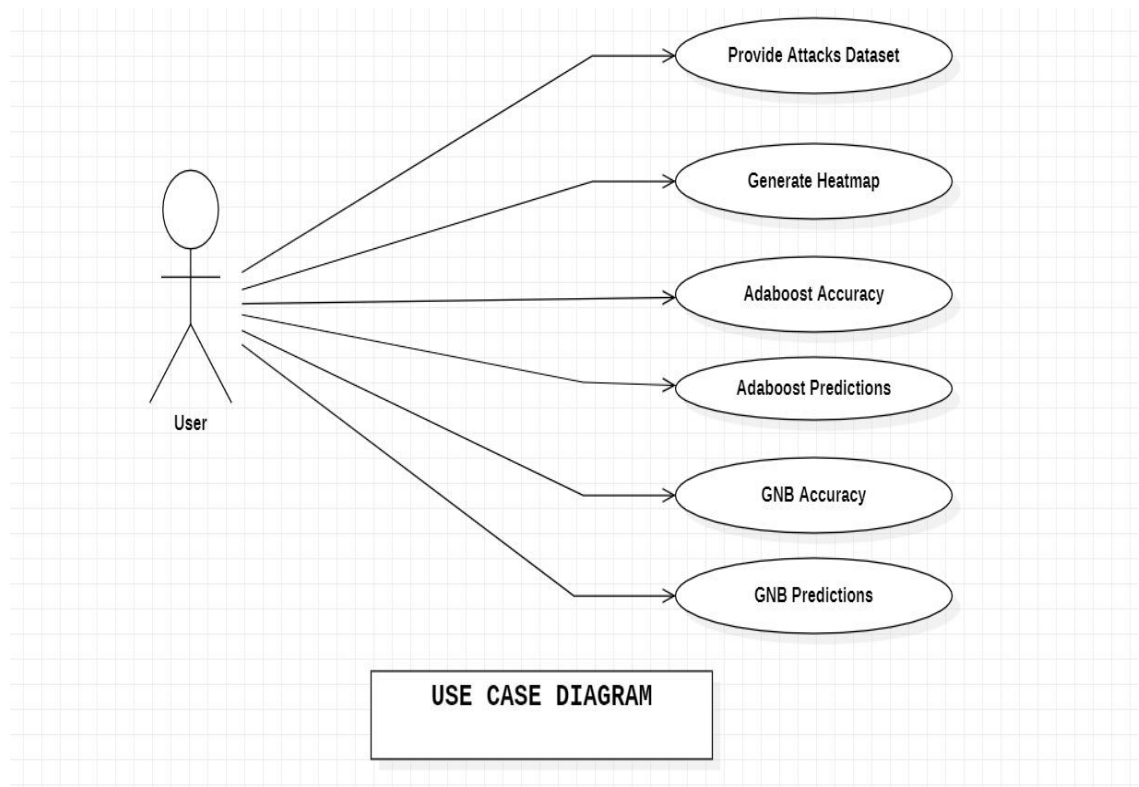
### 4.3.1 USE CASE DIAGRAM

A Use case is a description of set of sequence of actions. Graphically it is rendered as an ellipse with solid line including only its name. Use case diagram is a behavioral diagram that shows a set of use cases and actors and their relationship. It is an association between the use cases and actors. An actor represents a real-world object. Primary Actor – Sender, Secondary Actor Receiver.

Use case diagrams are a type of UML (Unified Modeling Language) diagram used in software engineering to describe the functional requirements of a system. They depict the various interactions between the system and its actors (users or other systems) by showing the different use cases or scenarios that the system supports.

Use case diagrams are used during the requirements gathering and analysis phase of software development to help understand the user's goals and requirements for the system. They provide a high-level view of the system's functionality, which can be used to guide the system's design and development

Use case diagrams can also be used to identify potential errors or ambiguities in the system's requirements before the development process begins.



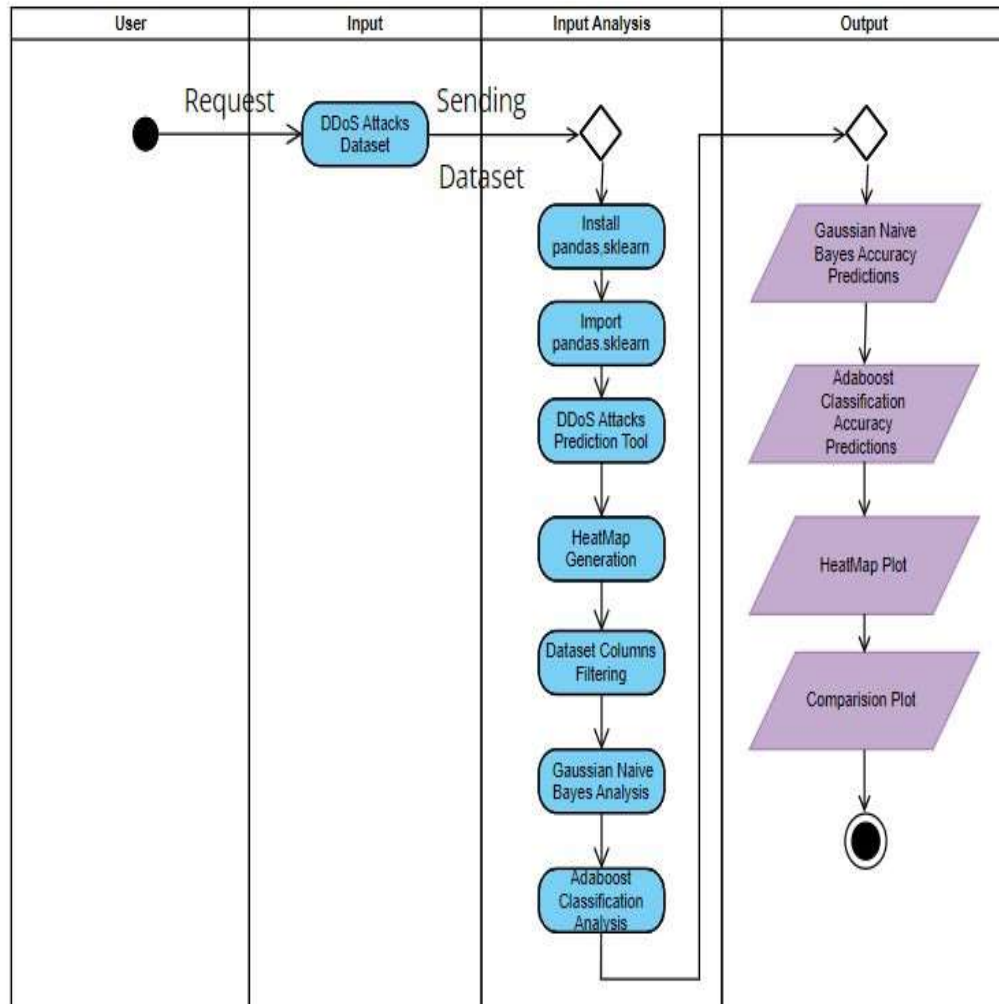
**Figure: 4.3 Use Case Diagram**

#### 4.4 Activity Diagram

Activity diagrams are a type of UML (Unified Modeling Language) diagram used in software engineering to describe the flow of activities or processes in a system. They provide a graphical representation of the steps involved in a particular activity or process, including the conditions, actions, and transitions between steps.

Activity diagrams are used during the design and implementation phase of software development to model and analyze the flow of activities or processes in the system. They can be used to identify potential bottlenecks, errors, or inefficiencies in the system's design, which can be addressed before the implementation process begins. Activity diagrams can also be used to document and communicate the system's behavior to stakeholders and other members of the development team.

Activity diagrams can also include swim lanes, which represent different organizational units or actors involved in the activity or process.



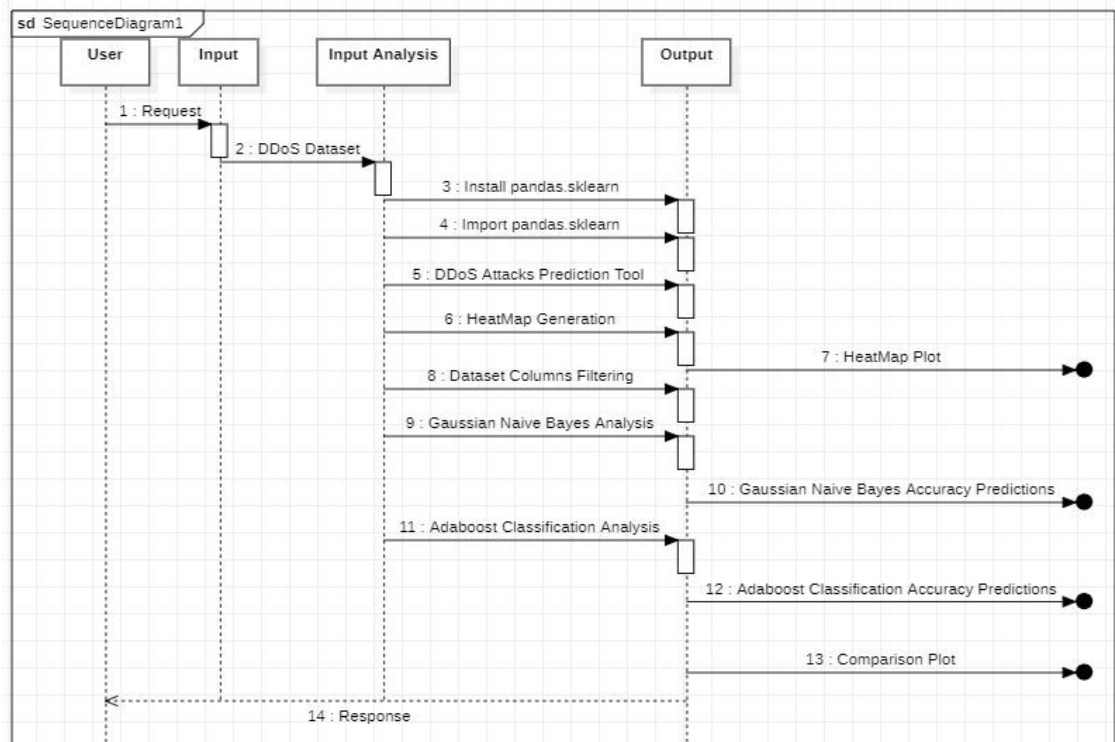
**Figure: 4.4 Activity Diagram**

## 4.5 Sequence Diagram

Sequence diagrams are a type of UML (Unified Modeling Language) diagram used in software engineering to describe the interactions between objects or components in a system. They provide a graphical representation of the sequence of messages exchanged between objects or components during a particular scenario or use case.

Sequence diagrams are used during the design and implementation phase of software development to model and analyze the interactions between objects or components in the system. They can be used to identify potential errors or ambiguities in the system's design, which can be addressed before the implementation process begins. Sequence diagrams can also be used to document and communicate the system's behavior to stakeholders and other members of the development team.

Sequence diagrams can also include conditions, loops, and other control structures to model more complex interactions between objects or components.



**Figure: 4.5 Sequence Diagram**

## 5.SAMPLE CODE

### 5.1 Coding

#### 5.1.1 gnb1.py:-

```
import os

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

import time

# reading features list

with open("kddcup.names", 'r') as f:

    f.read()

#Appending columns to the dataset and adding a new column name 'target' to
the dataset.

cols =""duration,

protocol_type,

service,

flag,

src_bytes,

dst_bytes,

land,

wrong_fragment,
```

urgent,  
  
hot,  
  
num\_failed\_logins,  
  
logged\_in,  
  
num\_compromised,  
  
root\_shell,  
  
su\_attempted,  
  
num\_root,  
  
num\_file\_creations,  
  
num\_shells,  
  
num\_access\_files,  
  
num\_outbound\_cmds,  
  
is\_host\_login,  
  
is\_guest\_login,  
  
count,  
  
srv\_count,  
  
error\_rate,  
  
srv\_error\_rate,  
  
rerror\_rate,  
  
srv\_rerror\_rate,  
  
same\_srv\_rate,  
  
diff\_srv\_rate,



```

srv_diff_host_rate,

dst_host_count,

dst_host_srv_count,

dst_host_same_srv_rate,

dst_host_diff_srv_rate,

dst_host_same_src_port_rate,

dst_host_srv_diff_host_rate,

dst_host_serror_rate,

dst_host_srv_serror_rate,

dst_host_rerror_rate,

dst_host_srv_rerror_rate"""

columns=[]

for c in cols.split(', '):

    if(c.strip()):

        columns.append(c.strip())

columns.append('target')

#print(len(columns))

#Reading the 'attack_types' file.

with open("training_attack_types", 'r') as f:

    f.read()

#Creating a dictionary of attack_types

attacks_types = {

```

'normal': 'normal',  
  
'back': 'dos',  
  
'buffer\_overflow': 'u2r',  
  
'ftp\_write': 'r2l',  
  
'guess\_passwd': 'r2l',  
  
'imap': 'r2l',  
  
'ipsweep': 'probe',  
  
'land': 'dos',  
  
'loadmodule': 'u2r',  
  
'multihop': 'r2l',  
  
'neptune': 'dos',  
  
'nmap': 'probe',  
  
'perl': 'u2r',  
  
'phf': 'r2l',  
  
'pod': 'dos',  
  
'portsweep': 'probe',  
  
'rootkit': 'u2r',  
  
'satan': 'probe',  
  
'smurf': 'dos',  
  
'spy': 'r2l',  
  
'teardrop': 'dos',  
  
'warezclient': 'r2l',

```

'warezmaster': 'r2l',

}

#Reading the dataset('kddcup.data_10_percent.gz') and adding

#Attack Type feature in the training dataset where attack type feature has 5
distinct values i.e. dos, normal, probe, r2l, u2r.

path = "kddcup.data_10_percent.gz"

df = pd.read_csv(path, names = columns)

# Adding Attack Type column

df['Attack Type'] = df.target.apply(lambda r:attacks_types[r[:-1]])

#print(df.head())

df.shape

#Finding missing values of all features

#print(df.isnull().sum())

#Finding Categorical Features

num_cols = df._get_numeric_data().columns

cate_cols = list(set(df.columns)-set(num_cols))

cate_cols.remove('target')

cate_cols.remove('Attack Type')

#print(cate_cols)

# Data Correlation – Find the highly correlated variables using heatmap and
ignore them for analysis

df = df.dropna('columns')# drop columns with NaN

```

```
df = df[[col for col in df if df[col].nunique() > 1]]# keep columns where there  
are more than 1 unique values
```

```
# This variable is highly correlated with num_compromised and should be  
ignored for analysis.
```

```
 #(Correlation = 0.9938277978738366)
```

```
df.drop('num_root', axis = 1, inplace = True)
```

```
# This variable is highly correlated with serror_rate and should be ignored for  
analysis.
```

```
 #(Correlation = 0.9983615072725952)
```

```
df.drop('srv_serror_rate', axis = 1, inplace = True)
```

```
# This variable is highly correlated with rerror_rate and should be ignored for  
analysis.
```

```
 #(Correlation = 0.9947309539817937)
```

```
df.drop('srv_rerror_rate', axis = 1, inplace = True)
```

```
# This variable is highly correlated with srv_serror_rate and should be ignored  
for analysis.
```

```
 #(Correlation = 0.9993041091850098)
```

```
df.drop('dst_host_srv_serror_rate', axis = 1, inplace = True)
```

```
# This variable is highly correlated with -+rerror_rate and should be ignored for  
analysis.
```

```
 #(Correlation = 0.9869947924956001)
```

```
df.drop('dst_host_serror_rate', axis = 1, inplace = True)
```

```
# This variable is highly correlated with srv_rerror_rate and should be ignored  
for analysis.
```

```
 #(Correlation = 0.9821663427308375)
```

```

df.drop('dst_host_error_rate', axis = 1, inplace = True)

# This variable is highly correlated with error_rate and should be ignored for
analysis.

#(Correlation = 0.9851995540751249)

df.drop('dst_host_srv_error_rate', axis = 1, inplace = True)

# This variable is highly correlated with srv_error_rate and should be ignored
for analysis.

#(Correlation = 0.9865705438845669)

df.drop('dst_host_same_srv_rate', axis = 1, inplace = True)

#Feature Mapping – Apply feature mapping on features such as :
‘protocol_type’ & ‘flag’

pmap = {'icmp':0, 'tcp':1, 'udp':2}

df['protocol_type'] = df['protocol_type'].map(pmap)

# flag feature mapping

fmap = {'SF':0, 'S0':1, 'REJ':2, 'RSTR':3, 'RSTO':4, 'SH':5, 'S1':6, 'S2':7,
'RSTOS0':8, 'S3':9, 'OTH':10}

df['flag'] = df['flag'].map(fmap)

#Remove irrelevant features such as ‘service’ before modelling

df.drop('service', axis = 1, inplace = True)

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import MinMaxScaler

# Splitting the dataset

df = df.drop(['target', ], axis = 1)

#print(df.shape)

```

```

df.to_csv('processed_data.csv')

# Target variable and train set

y = df[['Attack Type']]

X = df.drop(['Attack Type', ], axis = 1)

sc = MinMaxScaler()

X = sc.fit_transform(X)

# Split test and train data

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.33,
random_state = 42)

#print(X_train.shape, X_test.shape)

#print(y_train.shape, y_test.shape)

#Code: Python implementation of Guassian Naive Bayes

# Gaussian Naive Bayes

from sklearn.naive_bayes import GaussianNB

from sklearn.metrics import accuracy_score

clfg = GaussianNB()

clfg.fit(X_train, y_train.values.ravel())

print("Gaussian Naive Bayes score is:", clfg.score(X_train, y_train))

```

### 5.1.2 abc1.py:

```

import os

import pandas as pd

```

```

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

import time

# reading features list

with open("kddcup.names", 'r') as f:

    f.read()

#Appending columns to the dataset and adding a new column name 'target' to
the dataset.

cols = ""duration,

protocol_type,

service,

flag,

src_bytes,

dst_bytes,

land,

wrong_fragment,

urgent,

hot,

num_failed_logins,

logged_in,

num_compromised,

root_shell,

```

su\_attempted,  
  
num\_root,  
  
num\_file\_creations,  
  
num\_shells,  
  
num\_access\_files,  
  
num\_outbound\_cmds,  
  
is\_host\_login,  
  
is\_guest\_login,  
  
count,  
  
srv\_count,  
  
serror\_rate,  
  
srv\_serror\_rate,  
  
rerror\_rate,  
  
srv\_rerror\_rate,  
  
same\_srv\_rate,  
  
diff\_srv\_rate,  
  
srv\_diff\_host\_rate,  
  
dst\_host\_count,  
  
dst\_host\_srv\_count,  
  
dst\_host\_same\_srv\_rate,  
  
dst\_host\_diff\_srv\_rate,  
  
dst\_host\_same\_src\_port\_rate,



```

dst_host_srv_diff_host_rate,

dst_host_serror_rate,

dst_host_srv_serror_rate,

dst_host_rerror_rate,

dst_host_srv_rerror_rate"""

columns=[]

for c in cols.split(' '):

    if(c.strip()):

        columns.append(c.strip())

columns.append('target')

#print(len(columns))

#Reading the 'attack_types' file.

with open("training_attack_types", 'r') as f:

    f.read()

#Creating a dictionary of attack_types

attacks_types = {

    'normal': 'normal',

    'back': 'dos',

    'buffer_overflow': 'u2r',

    'ftp_write': 'r2l',

    'guess_passwd': 'r2l',

    'imap': 'r2l',

```

```
'ipsweep': 'probe',  
  
'land': 'dos',  
  
'loadmodule': 'u2r',  
  
'multihop': 'r2l',  
  
'neptune': 'dos',  
  
'nmap': 'probe',  
  
'perl': 'u2r',  
  
'phf': 'r2l',  
  
'pod': 'dos',  
  
'portsweep': 'probe',  
  
'rootkit': 'u2r',  
  
'satan': 'probe',  
  
'smurf': 'dos',  
  
'spy': 'r2l',  
  
'teardrop': 'dos',  
  
'warezclient': 'r2l',  
  
'warezmaster': 'r2l',  
  
}
```

```
#Reading the dataset('kddcup.data_10_percent.gz') and adding
```

```
#Attack Type feature in the training dataset where attack type feature has 5  
distinct values i.e. dos, normal, probe, r2l, u2r.
```

```
path = "kddcup.data_10_percent.gz"
```

```
df = pd.read_csv(path, names = columns)
```

```

# Adding Attack Type column

df['Attack Type'] = df.target.apply(lambda r:attacks_types[r[:-1]])

#print(df.head())

df.shape

#Finding missing values of all features

#print(df.isnull().sum())

#Finding Categorical Features

num_cols = df._get_numeric_data().columns

cate_cols = list(set(df.columns)-set(num_cols))

cate_cols.remove('target')

cate_cols.remove('Attack Type')

#print(cate_cols)

# Data Correlation – Find the highly correlated variables using heatmap and
ignore them for analysis

df = df.dropna('columns')# drop columns with NaN

df = df[[col for col in df if df[col].nunique() > 1]]# keep columns where there
are more than 1 unique values

# This variable is highly correlated with num_compromised and should be
ignored for analysis.

#(Correlation = 0.9938277978738366)

df.drop('num_root', axis = 1, inplace = True)

# This variable is highly correlated with serror_rate and should be ignored for
analysis.

```

```

#(Correlation = 0.9983615072725952)

df.drop('srv_error_rate', axis = 1, inplace = True)

# This variable is highly correlated with error_rate and should be ignored for
analysis.

#(Correlation = 0.9947309539817937)

df.drop('srv_error_rate', axis = 1, inplace = True)

# This variable is highly correlated with srv_error_rate and should be ignored
for analysis.

#(Correlation = 0.9993041091850098)

df.drop('dst_host_srv_error_rate', axis = 1, inplace = True)

# This variable is highly correlated with error_rate and should be ignored for
analysis.

#(Correlation = 0.9869947924956001)

df.drop('dst_host_error_rate', axis = 1, inplace = True)

# This variable is highly correlated with srv_error_rate and should be ignored
for analysis.

#(Correlation = 0.9821663427308375)

df.drop('dst_host_error_rate', axis = 1, inplace = True)

# This variable is highly correlated with error_rate and should be ignored for
analysis.

#(Correlation = 0.9851995540751249)

df.drop('dst_host_srv_error_rate', axis = 1, inplace = True)

# This variable is highly correlated with srv_error_rate and should be ignored
for analysis.

```

```

#(Correlation = 0.9865705438845669)

df.drop('dst_host_same_srv_rate', axis = 1, inplace = True)

#Feature Mapping – Apply feature mapping on features such as :
'protocol_type' & 'flag'

pmap = {'icmp':0, 'tcp':1, 'udp':2}

df['protocol_type'] = df['protocol_type'].map(pmap)

# flag feature mapping

fmap = {'SF':0, 'S0':1, 'REJ':2, 'RSTR':3, 'RSTO':4, 'SH':5, 'S1':6, 'S2':7,
'RSTOS0':8, 'S3':9, 'OTH':10}

df['flag'] = df['flag'].map(fmap)

#Remove irrelevant features such as 'service' before modelling

df.drop('service', axis = 1, inplace = True)

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import MinMaxScaler

# Splitting the dataset

df = df.drop(['target', ], axis = 1)

#print(df.shape)

df.to_csv('processed_data.csv')

# Target variable and train set

y = df[['Attack Type']]

X = df.drop(['Attack Type', ], axis = 1)

sc = MinMaxScaler()

X = sc.fit_transform(X)

```

```

# Split test and train data

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.33,
random_state = 42)

#print(X_train.shape, X_test.shape)

#print(y_train.shape, y_test.shape)

#Code: Python implementation of Guassian Naive Bayes

# Logistic Regression

from sklearn.ensemble import AdaBoostClassifier

from sklearn.metrics import accuracy_score

clf = AdaBoostClassifier()

clf.fit(X_train, y_train.values.ravel())

print("Adaboost Accuracy score is:", clf.score(X_train, y_train))

```

### 5.1.3 attacks1.py:

```

import sys

import os

from attacks import *

from PyQt5 import QtWidgets, QtGui, QtCore

class MyForm(QtWidgets.QMainWindow):

    def __init__(self, parent=None):

        QtWidgets.QWidget.__init__(self, parent)

        self.ui = Ui_MainWindow()

        self.ui.setupUi(self)

```

```

self.ui.pushButton_2.clicked.connect(self.dsetf)

self.ui.pushButton_7.clicked.connect(self.compare)

self.ui.pushButton_6.clicked.connect(self.abc1)

self.ui.pushButton_4.clicked.connect(self.gnb1)

self.ui.pushButton_5.clicked.connect(self.htmap)

def dsetf(self):

    os.system("python -W ignore attack2.py")

def compare(self):

    os.system("python compare1.py")

def abc1(self):

    os.system("python -W ignore abc1.py")

def gnb1(self):

    os.system("python -W ignore gnb1.py")

def htmap(self):

    os.system("python -W ignore attack1.py")

if __name__ == "__main__":

    app = QtWidgets.QApplication(sys.argv)

    myapp = MyForm()

    myapp.show()

    sys.exit(app.exec_())

```

#### 5.1.4 attack1.py:

```
import os

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

import time

# reading features list

with open("kddcup.names", 'r') as f:

    f.read()

#Appending columns to the dataset and adding a new column name 'target' to
the dataset.

cols = ""duration,

protocol_type,

service,

flag,

src_bytes,

dst_bytes,

land,

wrong_fragment,

urgent,
```



hot,  
  
num\_failed\_logins,  
  
logged\_in,  
  
num\_compromised,  
  
root\_shell,  
  
su\_attempted,  
  
num\_root,  
  
num\_file\_creations,  
  
num\_shells,  
  
num\_access\_files,  
  
num\_outbound\_cmds,  
  
is\_host\_login,  
  
is\_guest\_login,  
  
count,  
  
srv\_count,  
  
serror\_rate,  
  
srv\_serror\_rate,  
  
rerror\_rate,  
  
srv\_rerror\_rate,  
  
same\_srv\_rate,  
  
diff\_srv\_rate,  
  
srv\_diff\_host\_rate,

```

dst_host_count,

dst_host_srv_count,

dst_host_same_srv_rate,

dst_host_diff_srv_rate,

dst_host_same_src_port_rate,

dst_host_srv_diff_host_rate,

dst_host_error_rate,

dst_host_srv_error_rate,

dst_host_error_rate,

dst_host_srv_error_rate"""

columns=[]

for c in cols.split(', '):

    if(c.strip()):

        columns.append(c.strip())

columns.append('target')

#print(len(columns))

#Reading the 'attack_types' file.

with open("training_attack_types", 'r') as f:

    f.read()

#Creating a dictionary of attack_types

attacks_types = {

    'normal': 'normal',

```

'back': 'dos',  
  
'buffer\_overflow': 'u2r',  
  
'ftp\_write': 'r2l',  
  
'guess\_passwd': 'r2l',  
  
'imap': 'r2l',  
  
'ipsweep': 'probe',  
  
'land': 'dos',  
  
'loadmodule': 'u2r',  
  
'multihop': 'r2l',  
  
'neptune': 'dos',  
  
'nmap': 'probe',  
  
'perl': 'u2r',  
  
'phf': 'r2l',  
  
'pod': 'dos',  
  
'portsweep': 'probe',  
  
'rootkit': 'u2r',  
  
'satan': 'probe',  
  
'smurf': 'dos',  
  
'spy': 'r2l',  
  
'teardrop': 'dos',  
  
'warezclient': 'r2l',  
  
'warezmaster': 'r2l',

```

}

#Reading the dataset('kddcup.data_10_percent.gz') and adding

#Attack Type feature in the training dataset where attack type feature has 5
distinct values i.e. dos, normal, probe, r2l, u2r.

path = "kddcup.data_10_percent.gz"

df = pd.read_csv(path, names = columns)

# Adding Attack Type column

df['Attack Type'] = df.target.apply(lambda r:attacks_types[r[:-1]])

#print(df.head())

df.shape

#Finding missing values of all features

#print(df.isnull().sum())

#Finding Categorical Features

num_cols = df._get_numeric_data().columns

cate_cols = list(set(df.columns)-set(num_cols))

cate_cols.remove('target')

cate_cols.remove('Attack Type')

#print(cate_cols)

# Data Correlation – Find the highly correlated variables using heatmap and
ignore them for analysis

df = df.dropna('columns')# drop columns with NaN

df = df[[col for col in df if df[col].nunique() > 1]]# keep columns where there
are more than 1 unique values

```

```

corr = df.corr()

plt.figure(figsize=(15, 12))

sns.heatmap(corr)

plt.show()

##Note: The white colored boxes in the heat map, indicates the highly correlated
columns, and these columns can be dropped for further analysis

```

### 5.1.5 compare1.py:

```

import os

import numpy as np

import matplotlib.pyplot as plt

x = ['GNB', 'ABC']

y = [87.951, 40.03]

fig, ax = plt.subplots()

width = 0.50 # the width of the bars

ax.set_yticklabels(x, minor=False)

for i, v in enumerate(y):

    ax.text(v + 3, i + .25, str(v), color='blue', fontweight='bold')

plt.title('Gaussian Naive Bayes vs. Logistic Regression')

plt.xlabel('x')

plt.ylabel('y')

plt.show()

# plt.savefig(os.path.join('test.png'), dpi=300, format='png', bbox_inches='tight')
# use format='svg' or 'pdf' for vectorial pictures

```

## 6.TESTING

### 6.1 Software Testing

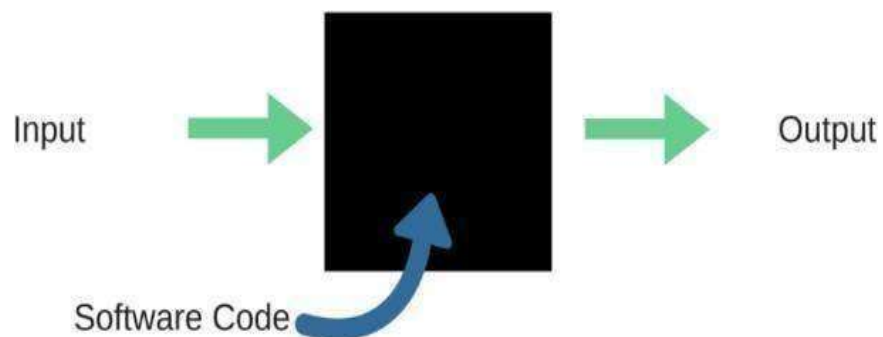
Software testing is a critical element of software quality assurance and represents the ultimate reviews of specification, design and coding. Testing represents an interesting anomaly for the software. During earlier definition and development phases, it was attempted to build software from an abstract concept to a tangible implementation. No system is error free because it is so till the next error crops up during any phase of the development or usage of the product. A sincere effort however needs to be put to bring out a product that is satisfactory.

Testing is a process of executing a program with the aim of finding error. To make our software perform well it should be error free. If testing is done successfully, it will remove all the errors from the software.

#### 6.1.1 Types of Testing

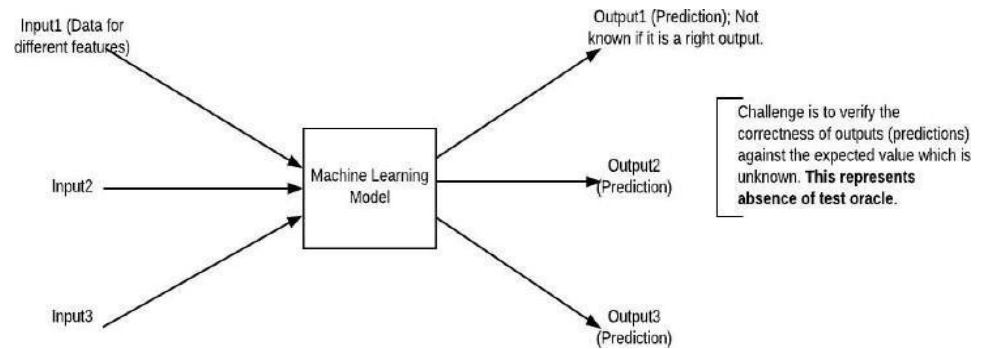
1. **Unit Testing:** Software verification and validation method in which a programmer tests if individual units of source code fit for use. It is usually conducted by the development team.
2. **Integration Testing:** The phase in software testing in which individual software modules are combined and tested as a group. It is usually conducted by testing teams.
3. **Alpha Testing:** Type of testing a software product or system conducted at the developer's site. Usually, it is performed by the end users.
4. **Beta Testing:** Final testing before releasing application for commercial purpose. It is typically done by end-users or others.

5. **Performance Testing:** Functional testing conducted to evaluate the compliance of a system or component with specified performance requirements. It is usually conducted by the performance engineer.
6. **White Box Testing:** This technique based on knowledge of the internal logic of an application's code and includes tests like coverage of code statements, branches, paths, conditions. It is performed by software developers.
7. **Black Box Testing:** A method of software testing that verifies the functionality of an application without knowing the details of its implementation including internal program, data structures etc. Test cases for black-box testing are created based on the requirement specifications. Therefore, it is also called as specification-based testing.



**Figure 6.1.1 Black Box Testing**

When applied to machine learning models, black box testing would mean testing machine learning models without knowing the internal details such as features of the machine learning model, the algorithm used to create the model etc.



**Figure 6.1.2 Black Box Testing for Machine Learning**

Test Case 1	
<b>Test Case Name</b>	Data Columns Filtering
<b>Description</b>	The co-related columns are dropped to reduce the unnecessary load on dataset.
<b>Output</b>	The processed CSV file is generated.
<b>Status</b>	Pass

Test Case 2	
<b>Test Case Name</b>	Gaussian Naive Bayes Analysis
<b>Description</b>	GNB Analysis is used to get the optimal accuracy score.
<b>Output</b>	GNB accuracy score is generated in the Command Prompt.
<b>Status</b>	Pass



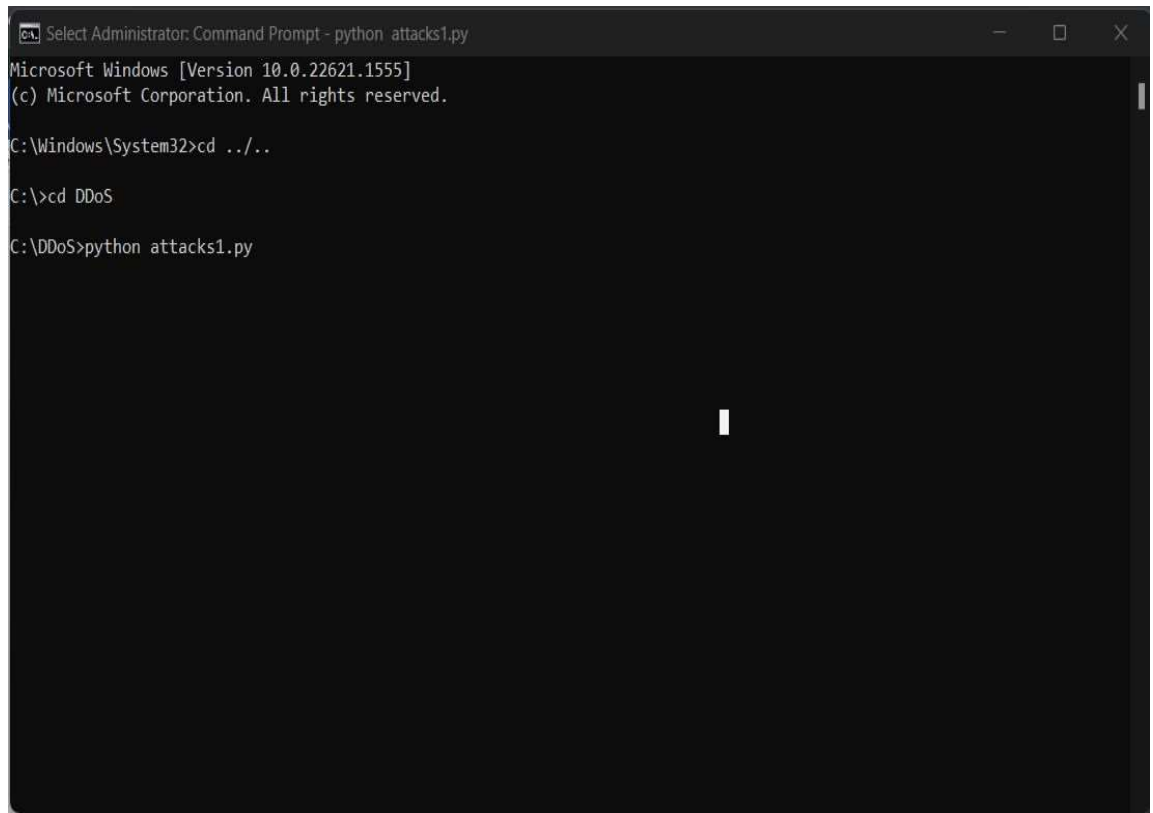
<b>Test Case 3</b>	
<b>Test Case Name</b>	Adaboost Analysis
<b>Description</b>	Adaboost Classifier is also used to get ideal accuracy score.
<b>Output</b>	Adaboost accuracy score is generated in the Command Prompt.
<b>Status</b>	Pass

<b>Test Case 4</b>	
<b>Test Case Name</b>	Comparison Plot
<b>Description</b>	The graph between GNB and ABC accuracy score is plotted.
<b>Output</b>	The graph is displayed in the GUI.
<b>Status</b>	Pass

<b>Test Case 5</b>	
<b>Test Case Name</b>	Heat Map Generation
<b>Description</b>	The co-related columns are identified by generating heatmap.
<b>Output</b>	Heat Map plot is displayed in the GUI.
<b>Status</b>	Pass

## 7.OUTPUT SCREENS

### 7.1 Running The Anaconda Prompt



```

Select Administrator: Command Prompt - python attacks1.py
Microsoft Windows [Version 10.0.22621.1555]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>cd ../../

C:\>cd DDoS

C:\DDoS>python attacks1.py

```

**Figure: 7.1 Command to enter into GUI**

## 7.2 Home Screen

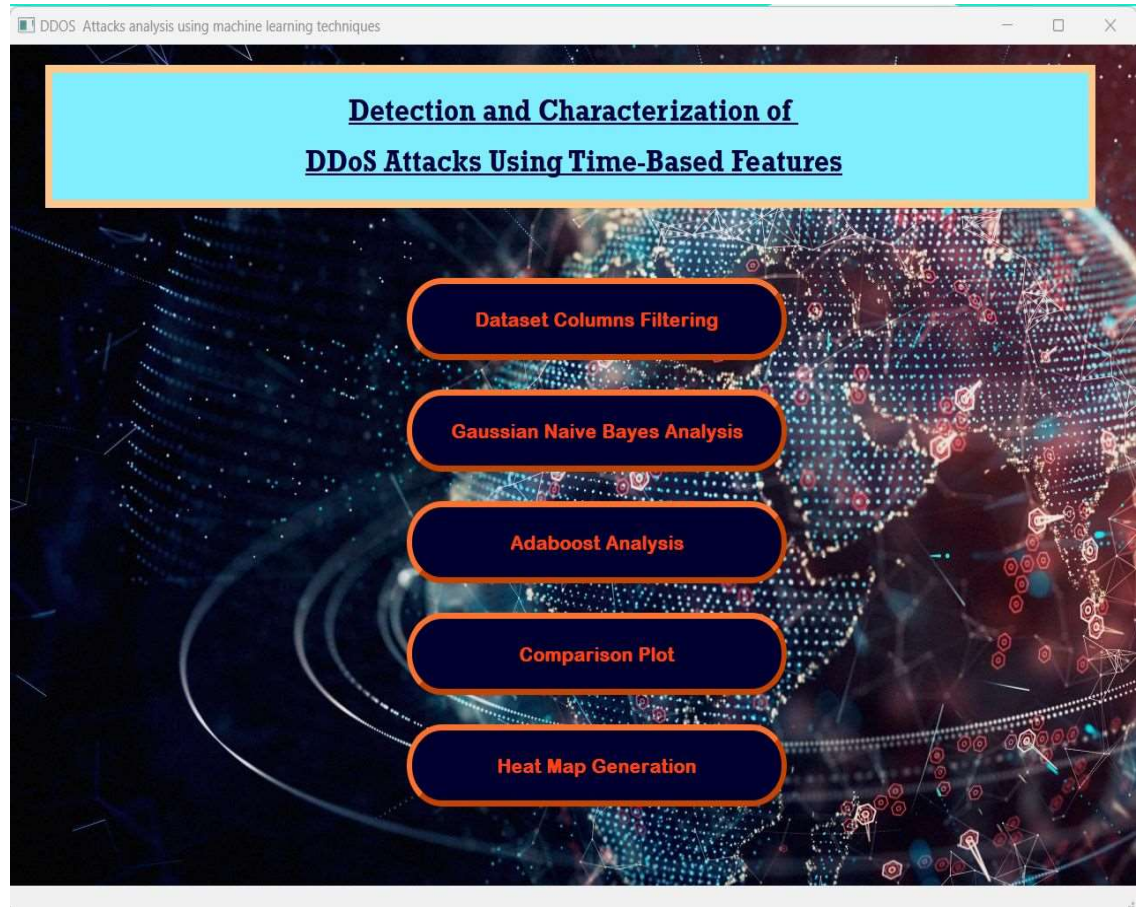


Figure: 7.2 Home screen GUI

### 7.3 Data Columns Filtering

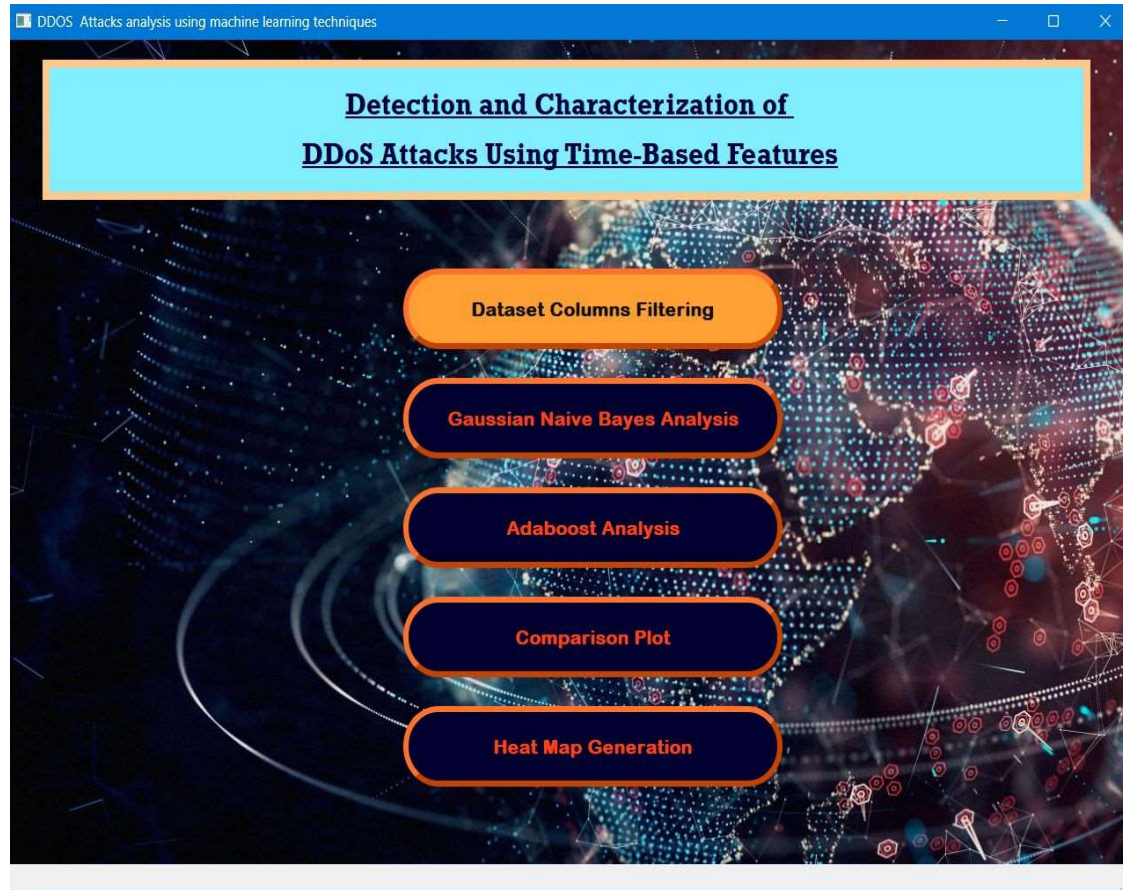
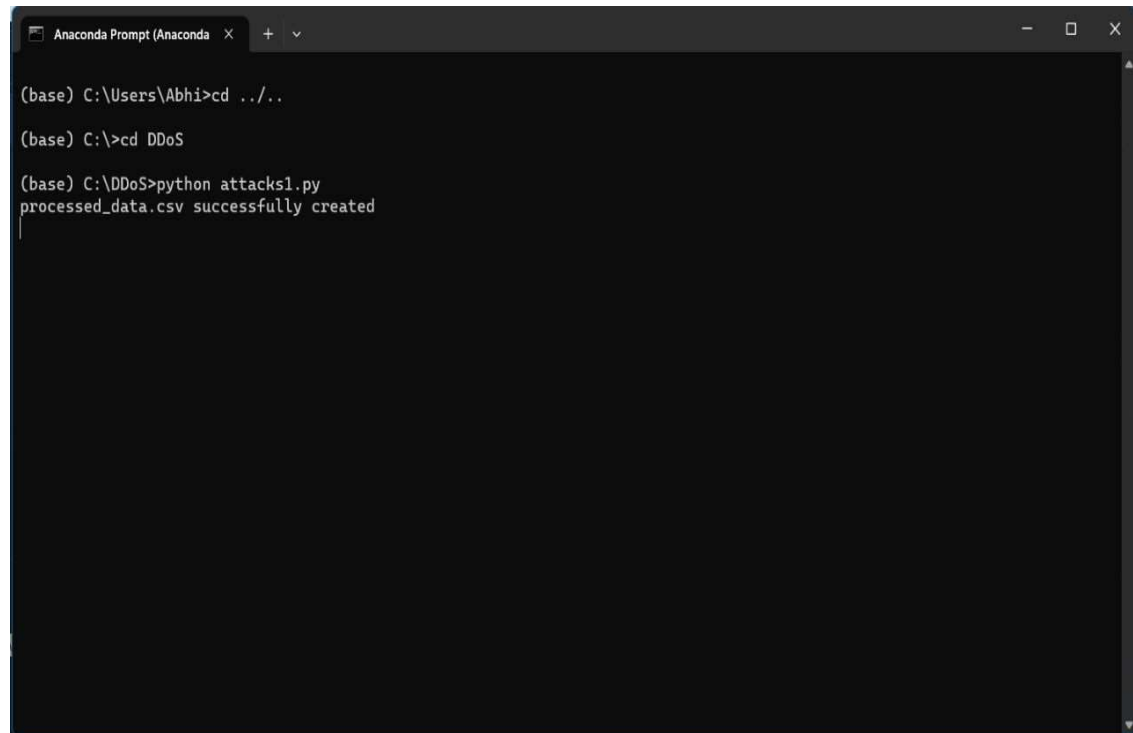


Figure: 7.3 Data columns filtering button is selection

## 7.4 CSV File



```
Anaconda Prompt (Anaconda) x + v
(base) C:\Users\Abhi>cd ../../
(base) C:\>cd DDoS
(base) C:\DDoS>python attacks1.py
processed_data.csv successfully created
```

**Figure: 7.4 Processed .csv file created**

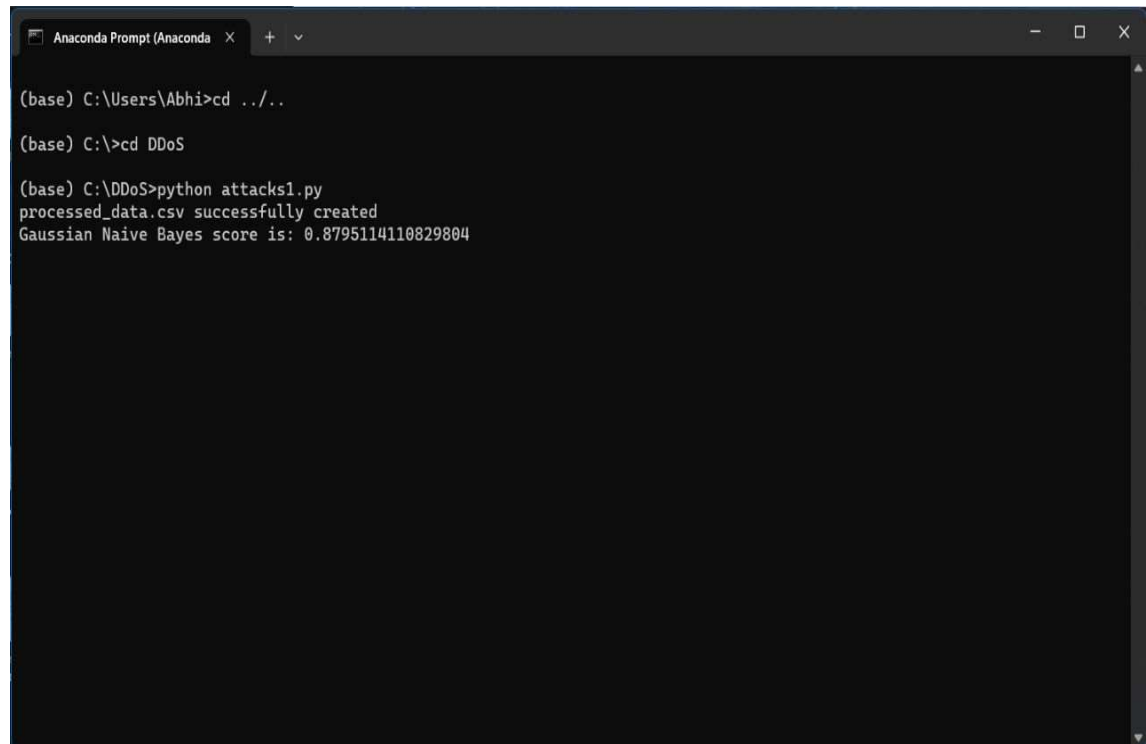
## 7.5 Gaussian Naive Bayes Analysis



Figure: 7.5 GNB analysis button is selection



## 7.6 Accuracy Score of GNB

A screenshot of an Anaconda Prompt terminal window. The window has a dark background and a title bar that says "Anaconda Prompt (Anaconda)". The terminal shows the following commands and output:

```
(base) C:\Users\Abhi>cd ../../  
(base) C:\>cd DDoS  
(base) C:\DDoS>python attacks1.py  
processed_data.csv successfully created  
Gaussian Naive Bayes score is: 0.8795114110829804
```

**Figure: 7.6 GNB accuracy score is generated**

## 7.7 Adaboost Analysis

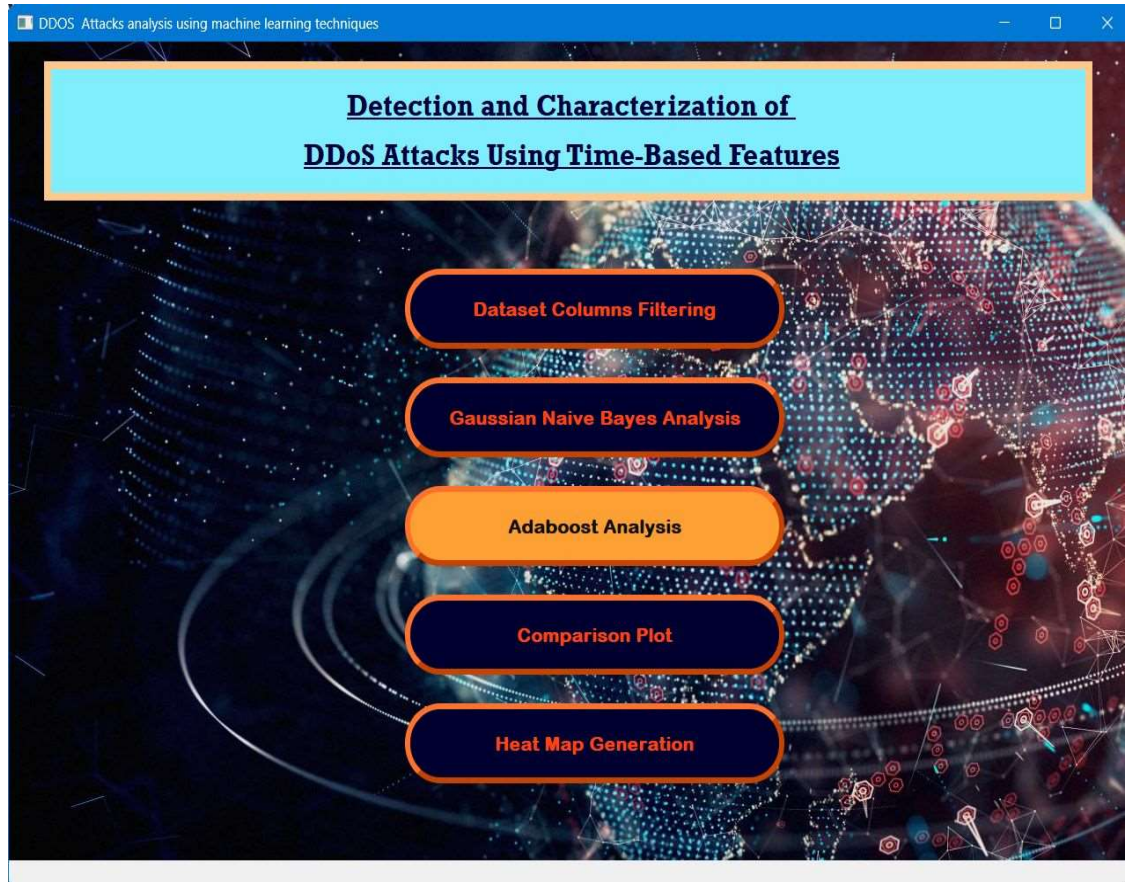
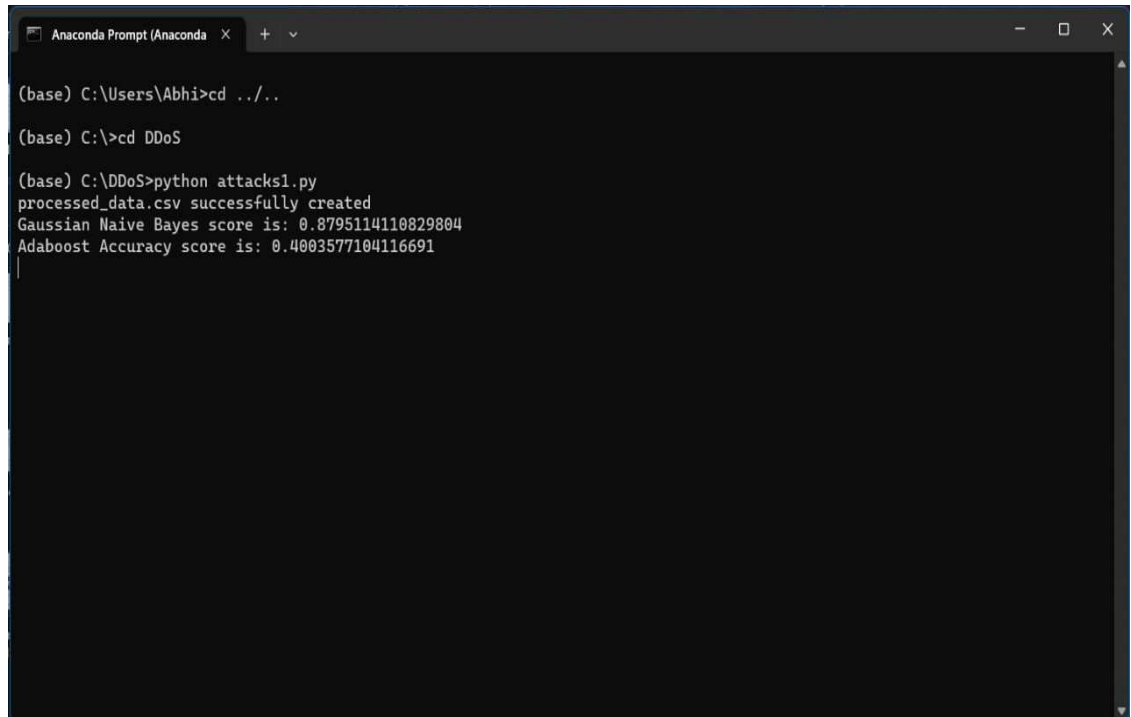


Figure: 7.7 ABC analysis button is selection



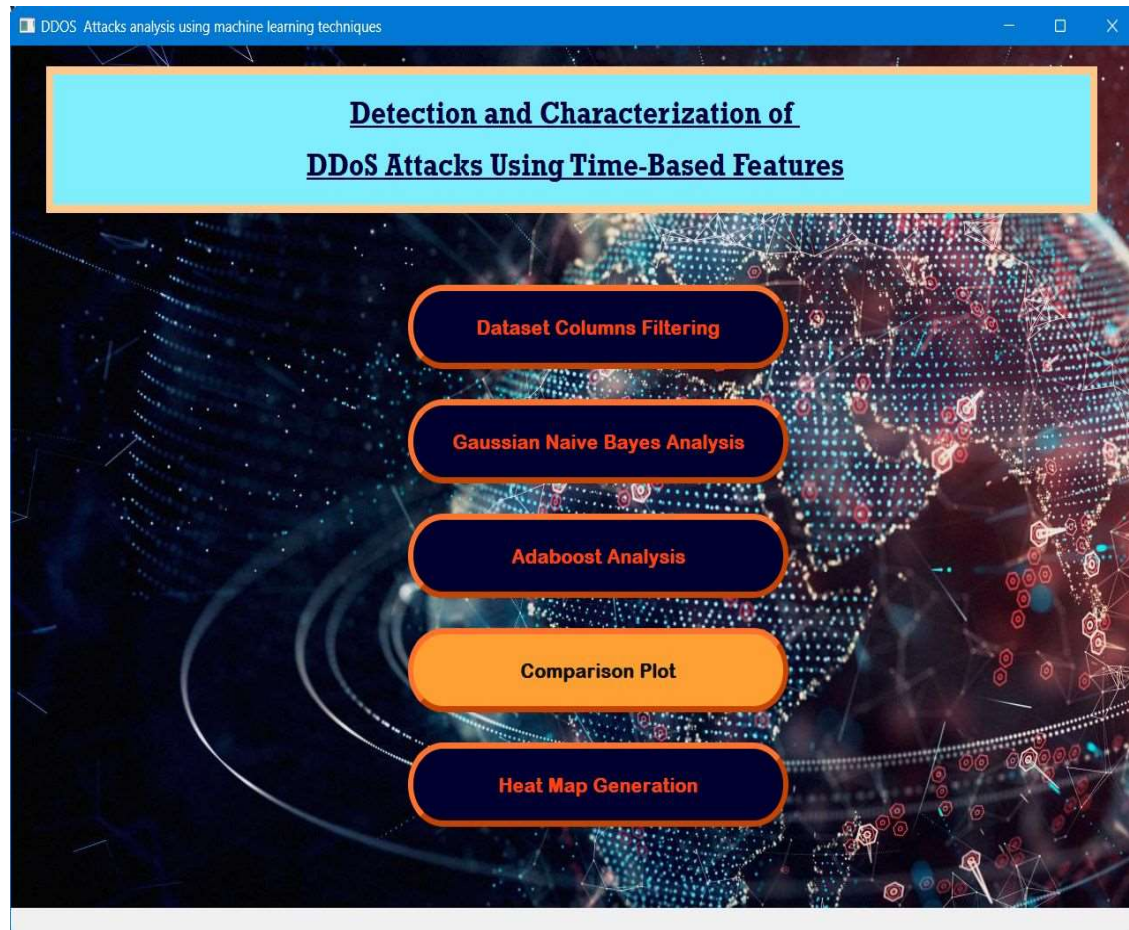
## 7.8 Accuracy Score of ABC



```
Anaconda Prompt (Anaconda) x + v
(base) C:\Users\Abhi>cd ../../
(base) C:\>cd DDoS
(base) C:\DDoS>python attacks1.py
processed_data.csv successfully created
Gaussian Naive Bayes score is: 0.8795114110829804
Adaboost Accuracy score is: 0.4003577104116691
```

**Figure: 7.8 ABC accuracy score is generated**

## 7.9 Comparison Plot



**Figure: 7.9 Comparison plot button is selected**

## 7.10 Comparison Graph

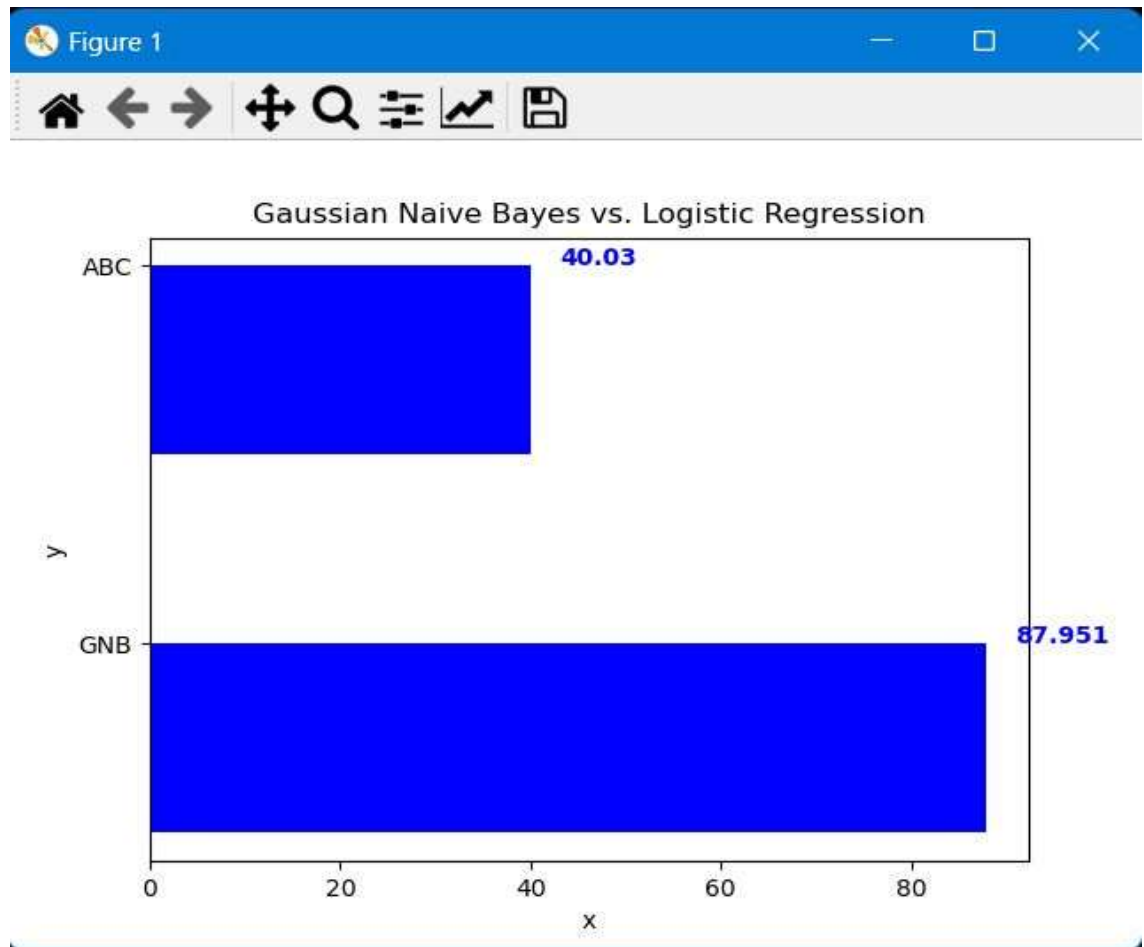


Figure: 7.10 Comparison graph of GNB and ABC

## 7.11 Heatmap Generation

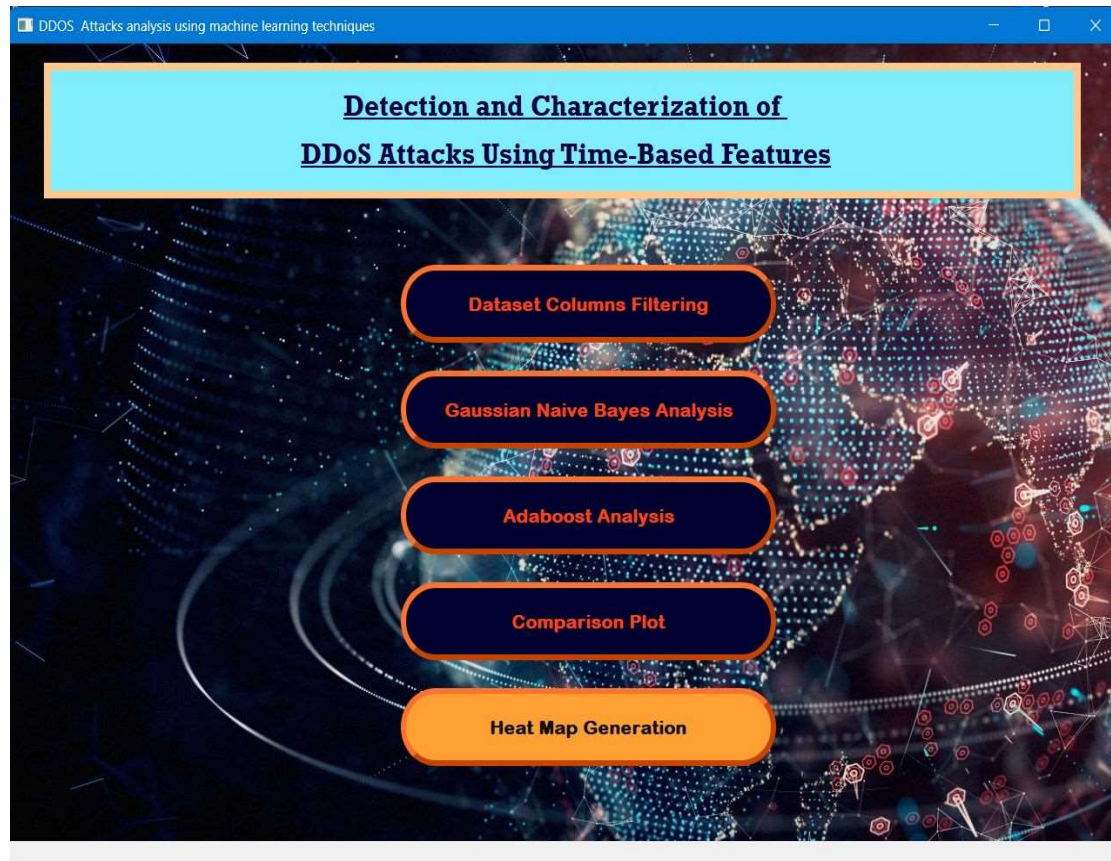
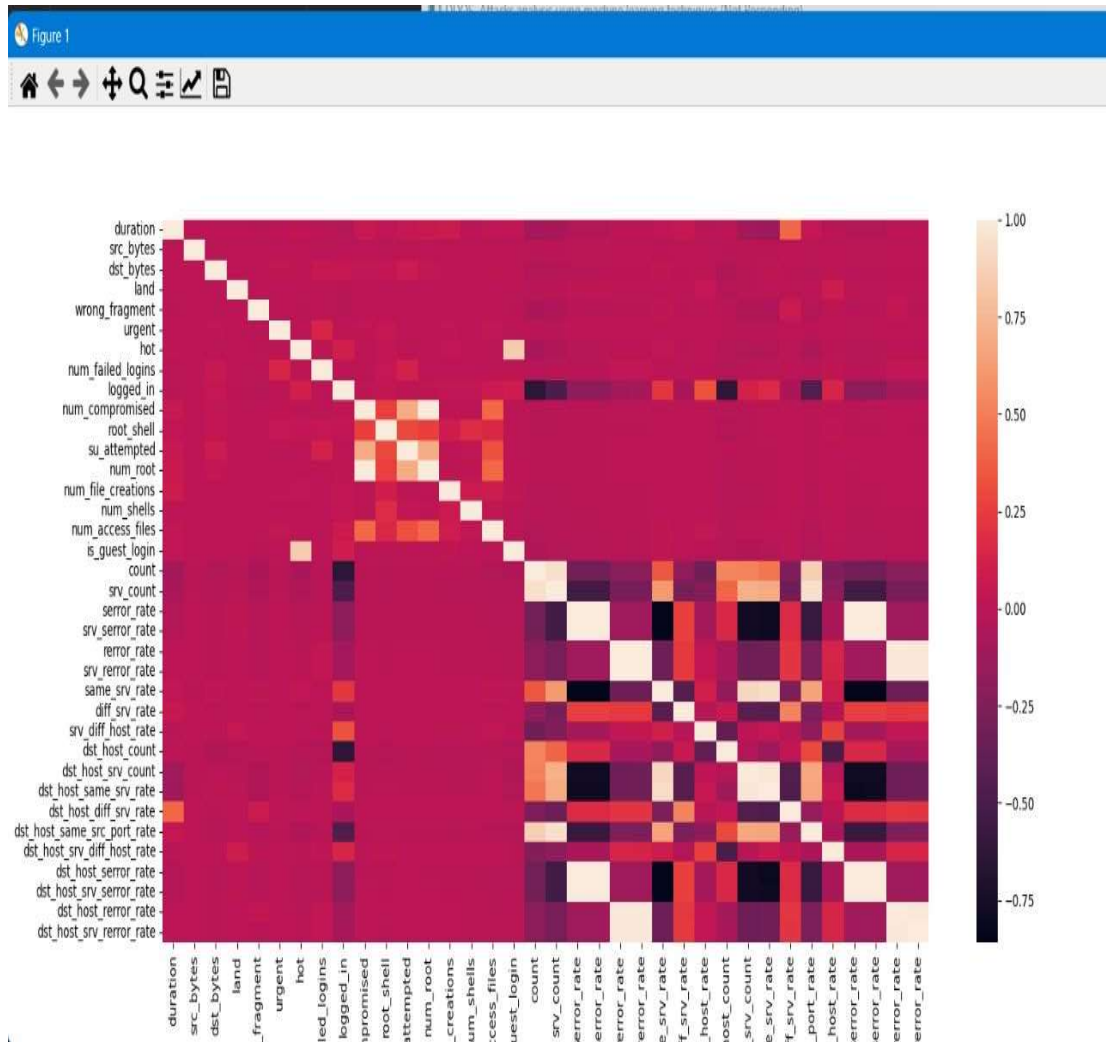


Figure: 7.11 Heatmap button is selected

## 7.12 Heatmap Graph



**Figure: 7.12 Heatmap plot is generated**

## **8.CONCLUSION**

### **8.1 Conclusion**

The project "DDoS Attacks analysis using machine learning techniques" is a valuable contribution to the field of cybersecurity. By using Gaussian Naive Bayes and Adaboost Classification methods, the project provides a reliable and accurate means of classifying cyber-attacks. Moreover, the project allows data analysts to gain a better understanding of heat maps and comparison plots, which can help them make informed decisions when it comes to securing network connections.

Overall, the project has the potential to improve the quality of secured network connections and reduce the risk of DDoS attacks. It can be a useful tool for organizations to predict and prevent potential DDoS attacks, which can save them a significant amount of time and resources that would have been otherwise spent on dealing with the aftermath of an attack.

Furthermore, this project has opened up opportunities for further research in the field of cybersecurity. Machine learning techniques can be applied to other areas of cybersecurity to improve the accuracy and reliability of existing systems. In the future, the application of these techniques could lead to the development of new cybersecurity solutions that are more efficient and effective in protecting against cyber threats.

Overall, this project has provided valuable insights into the use of machine learning techniques for analyzing DDoS attacks, and its findings can be used to develop better cybersecurity strategies in the future.

## 8.2 Future Enhancements

Some potential future enhancements of the above project could include:

- a. Integration with real-time network traffic:** Currently, the project analyzes a pre-collected dataset of DDoS attacks. To make it more useful in practical scenarios, the project could be enhanced to work with real-time network traffic and identify potential DDoS attacks in real-time.
- b. Multiple machine learning algorithms:** While the project uses Gaussian Naive Bayes and Adaboost Classification, there are many other machine learning algorithms that can be used for classification tasks. Future enhancements could involve the exploration of other algorithms and comparing their results.
- c. Deep learning:** Deep learning techniques, such as convolutional neural networks and recurrent neural networks, have shown promising results in detecting cyber-attacks. Future enhancements could involve exploring these techniques and comparing their performance with the existing methods.
- d. User interface improvements:** The current project has a basic user interface, and future enhancements could involve improving the user interface to make it more intuitive and user-friendly.
- e. Integration with security systems:** The project could be integrated with existing security systems to trigger automatic responses to potential DDoS attacks. This could include blocking IP addresses or limiting traffic from certain sources.
- f. Identifying new types of DDoS attacks:** As attackers continue to develop new techniques for DDoS attacks, the project could be enhanced to identify these new types of attacks and develop countermeasures.

## 9. BIBILOGRAPHY

### 9.1 Book References

1. JAMES HALLADAY, DRAKE CULLEN, NATHAN BRINER, JACKSON WARREN, KARSON FYE, RAM BASNET, JEREMY BERGEN AND TENZIN DOLECK: Detection and Characterization of DDoS Attacks Using Time-Based Features, IEEE Access.
2. T. Mahjabin, Y. Xiao, G. Sun, and W. Jiang:"A survey of distributed denial-of service attack, prevention, and mitigation techniques",Int. J. Distrib. Sensor Netw., vol. 13, no. 12, Dec. 2017, Art. no. 155014771774146.
3. Alparslan, O.; Gunes, O.; Hanay, Y.S.; Arakawa, S.I.; Murata, M. Improving resiliency against DDoS attacks by SDN and multipath orchestration of VNF services. In Proceedings of the 2017 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN), Osaka, Japan, 12–14 June 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1–3.
4. Bhatia, S. Ensemble-based model for DDoS attack detection and flash event separation. In Proceedings of the 2016 Future Technologies Conference (FTC), San Francisco, CA, USA, 6–7 December 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 958–967.
5. Chen, C.; Chen, H. A resource utilization measurement detection against DDoS attacks. In Proceedings of the 2016 9<sup>th</sup> International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), Datong, China, 15–17 October 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 1938–1943.
6. Jazi, H.H.; Gonzalez, H.; Stakhanova, N.; Ali, A. Ghorbani. Detecting HTTP-based application layer DoS attacks on web servers in the presence of sampling. Comput. Netw. 2017, 121, 25–36. [CrossRef]



## **9.2 Web References**

### **9.2.1 For Python**

<https://www.python.org/>

### **9.2.2 For Qt Designer**

<https://doc.qt.io/qt-5/qtdesigner-manual.html>

### **9.2.3 For SQLite**

<https://www.sqlite.org/about.html>

### **9.2.4 For PyQt**

<https://riverbankcomputing.com/software/pyqt/>

### **9.2.5 For Scikit-learn**

<https://scikit-learn.org/stable/>

### **9.2.6 For Matplotlib**

<https://matplotlib.org/>

## **10. APPENDICES**

### **10.1 Software Used**

#### **10.1.1 Python**

Python is an interpreted, high-level programming language that was first released in 1991. It is designed to be easy to learn and read, which makes it a popular choice for beginners and experienced programmers alike. Python is used for a wide range of applications, including scientific computing, data analysis, web development, machine learning, and more. Some of the key features of Python include its simple syntax, dynamic typing, and built-in support for common programming tasks such as file I/O and regular expressions.

#### **10.1.2 PyQt Designer**

PyQt Designer is a visual interface design tool that is used to create user interfaces for PyQt applications. It is a drag-and-drop interface that allows you to place widgets and other UI elements on a form, and then customize their properties using a properties editor. PyQt Designer supports a wide range of widgets and layout managers, making it easy to create complex and professional-looking interfaces. Once you have designed your interface, you can save it as a .ui file, which can be converted to Python code using Pyuic.

### **10.1.3 SQLite3**

SQLite3 is a lightweight, file-based relational database management system that is used to store and retrieve data. It is often used in small-scale applications and mobile apps because of its simplicity and ease of use. SQLite3 stores data in a single file, which makes it easy to distribute and deploy. It supports a wide range of data types, including text, integer, real, blob, and null. SQLite3 is also very fast and efficient, which makes it a good choice for applications that need to work with large amounts of data.

### **10.1.4 Pyuic**

Pyuic is a command-line tool that is used to convert PyQt Designer .ui files to Python code. It is a utility tool that comes with the PyQt library, and it takes the .ui file created in PyQt Designer and generates the corresponding Python code. The generated code can be used directly in the application, which saves time and effort in coding the user interface. Pyuic supports a wide range of PyQt widgets and layout managers, making it easy to convert even complex interfaces to Python code.

## **10.2 Testing Methods**

### **10.2.1 Key Board/Mouse**

Using the keyboard and mouse commands are given by the user and the user can access the command prompt. The user can basically use mouse to navigate through GUI, select the appropriate button and get the outputs accordingly.

## 11. PLAGIARISM REPORT

**PapersOwl**

Services ▾ Writing Tools ▾ How it Works Support About us ▾ [→ LOGIN](#) [ORDER NOW](#)

### Free Online Plagiarism Checker

```
selfuiobutton2clickedconnectselfsetf selfuiobutton7clickedconnectselfcompare
selfuiobutton6clickedconnectselfabc1 selfuiobutton4clickedconnectselfgnb1
selfuiobutton5clickedconnectselfhtmap def dself self ossystempython -w ignore attack2py def
compareself ossystempython comparetpy def abctself ossystempython -w ignore abctpy def
gnbtself ossystempython -w ignore gnbtpy def htmapself ossystempython -w ignore attack1py if
name main:
    app QtUiDateApplication
    curRow = 0
    muForm = muForm
    muForm.show()
    curRow += 1
```

45 words (518 characters) [↻ Recheck this text after changes](#) [📄 Check another text](#)

SIMILAR	ORIGINAL
8.0% 	92.0%
Well done, your text is unique!	

Need an essay written but don't have the time?

With PapersOwl you'll get it professionally researched,  
written and received right on time!

[GET MY ESSAY DONE](#)

**Figure: 11 Plagiarism Report**